Payroll System for Premier Instruments and Controls Ltd. (PRICOL)

PROJECT REPORT

Submitted by

J. DEEPA
Reg. No. 9338M0187

Under the Guidance of
Prof. P. SHANMUGAM, M.Sc.,(Engg),M.S.(Hawai)M.I.S.T.E.,
Prof. and H. O. D. of Computer Science and Engineering

In partial fulfilment of the requirements
for the award of the Degree of
MASTER OF COMPUTER APPLICATIONS
in the Department of Computer Science and Engineering of
Kumaraguru College of Technology
Affiliated to Bharathiar University, Coimbatore

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE - 641 006

JUNE 1996

CERTIFICATE

This is to certify that this project work entitled **PAYROLL ~~SYSTEM~~** Processing **FOR PREMIER INSTRUMENTS AND CONTROLS LTD (PRICOL)** [illegible] submitted to Kumaraguru College of Technology, Coimbatore (affiliated to Bharathiar University) in partial fulfilment of the requirements for the award of the Degree of **Master Of Computer ~~Applications~~** B.Sc [illegible] Computer Technology) is a record of original work done by ~~**DEEPA.**~~ **J,** SUNDAR C. SIVARAM V. ANIVERAN [illegible] (Reg. No ~~**9338M0187**~~ [illegible]) during her period of study in the Department of Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore under my supervision and guidance and this project work has not formed the basis for the award of any Degree / Diploma / Associateship / Fellowship or similar title to any candidate of any University.

Professor & Head

Staff in Charge

Submitted for University Examination held on ----11/4/96--------.

Internal Examiner

External Examiner

P-246

1st June, 1996

CERTIFICATE

This is to certify that Ms. J. Deepa, a final year (B.Sc. M.C.A. student of Kumaraguru College Of Technology, Coimbatore has successfully completed the project work titled 'PAYROLL SYSTEM FOR PREMIER UMA MAHESHWARI [illegible] INSTRUMENTS AND CONTROLS LTD (PRICOL)', a client concern of our organization. She has done the project from 2nd January 1996 to 31st May 1996.

The system was developed using SYBASE RDBMS under HP-UX 9.04 Operating System. During her association with us, she was found to be very good and sincere in her assignment.

UMA.

For PRIME TECHNOLOGIES,

Shankar

(D. SHANKAR)

Managing Director

ACKNOWLEDGEMENTS

I express my profound gratitude to **Dr.S.Subramaniam, M.Sc.(Engg), Ph.D., S.M.IEEE., M.I.S.T.E., Principal, Kumaraguru College Of Technology** for having inspired me to do this project work and providing me with the necessary facilities.

I express my hearty thanks to Mrs Chitradevi ~~**Prof.P.Shanmugam,** M.Sc.(Engg)~~, M.S.(Hawai), M.I.S.T.E., Lecturer ~~**Professor and**~~ **Head** of the Department of Computer **Science and Engineering**, who is also my Project **Guide**, for his valuable suggestions, guidance and support throughout the course of this project.

My sincere thanks to **Mr.D.Shankar, B.E., Managing Director,** [illegible] ~~**PRIME TECHNOLOGIES**~~, Coimbatore for providing us ~~me~~ with this great opportunity to do this project, for his extensive training and for having provided us ~~me~~ with all the resources.

I extend my thanks to all the faculty of the Department of Computer Science and Engineering for their able guidance and support during the course of this project.

SYNOPSIS

The project work was undertaken at a fast expanding software consultancy **M/S PRIME TECHNOLOGIES**, Coimbatore. The Payroll System was developed for their client **PREMIER INSTRUMENTS AND CONTROLS LTD** (PRICOL), Coimbatore. It is an ISO 9000 certified company with about 1687 employees working currently. The existing Payroll System was developed by HCL using an RDBMS called GENESIS. The system had various drawbacks. A period of 30 days was spent for the study of the existing system and a new system overcoming the existing drawbacks was designed. It took four months time to complete the coding and implementation work. On the date of submission of this project work the new Payroll System is in use at PRICOL.

The Payroll System has various modules such as Attendance Maintenance, Employee Information System, Variable Earnings Maintenance, Variable Deductions Maintenance (loans and advances), Pay Processing, Payslip Printing on pre-printed stationary, several reports and salary statements for the Management. It also has a provision for backups.

The new Payroll System was developed in a multiuser environment on a HP 9000 series 817 machine with HP-UX Ver 9.04 as the operating system. The RDBMS used was SYBASE (Ver 4.00). Various tools provided by SYBASE for application development such as ISQL (Interactive Structured Query Language), APT (Application Productivity Tool) Workbench, APT-Library, DB-Library, DWB (Data Workbench), etc., were used. The existing data in GENESIS was converted to SYBASE format using the 'bcp' (Bulk Copy) option provided by SYBASE.

CONTENTS

INTRODUCTION

THE IMPORTANCE OF PAYROLL SYSTEM

In an organization where hundreds or thousands of people are employed, a considerable portion of the expenditure of the business is on the compensation given to its employees. It is of prime importance that an accurate account of the compensation be maintained for use in maintaining all financial accounts of the company. The Payroll department maintains a record of the jobs, departments of the employees and the monetary reimbursements given to them. It records the total time spent by each individual on his job and accordingly computes his earnings. It prepares the payroll for each department showing the total amount earned for the period by each employee, any deductions made from his salary, any savings made by him and finally disburses the money to the employees.

For the management, the payroll department produces regular reports which makes them aware of the total expenditure incurred on compensation for employees, department-wise expenditure and numerous other details. So one can imagine how to manage the journey on choppy waters without a Payroll System.

A company may have many types of employees doing many types of tasks. These employees can be divided into full time and part time workers. They could be earning monthly salaries, hourly wages and getting overtime wages. Each of these variables needs to be incorporated into the basic design of a payroll system.

FUNCTIONS OF PAYROLL SYSTEM

One of the most important functions of the Payroll System is to generate the Payslip. Some of the details that are commonly present in a payslip are as follows.

Depending on the job status of an employee, a salaried employee receives a fixed amount as compensation called the Basic Pay. Most of the other additional compensations are calculated as a percentage of the basic pay.

One of these additionals is the Dearness Allowance (DA). The economy has a high rate of inflation. Due to constantly rising prices, the 'cost of living' is high. Dearness Allowance is paid depending on the cost of living to make up for inflation or rising prices. The higher the basic pay, lower the rate of DA, since a person getting a higher salary is anyway in a better position to face the rising prices.

Other incentives include House Rent Allowance (HRA), Leave Travel Allowance (LTA), Conveyance Allowance (CA) and the City Compensatory Allowance (CCA). Based on the company's policy, the employees will receive compensation for family accommodation. HRA is calculated as a percentage of the Basic pay. Leave Travel Allowance is a compensation paid to the employees towards expenditure incurred on a vacation. The compensation may be paid in full or partially for the expenditure incurred. The Conveyance Allowance is given to the employees to cover the expenses they incur in commuting to and from the office. The City Compensatory Allowance tries to compensate for the higher cost of living in certain parts of the country.

Over and above these fringe benefits, employees are given a bonus once every year which signifies their share in the profits earned by the company. It is again calculated as a percentage of their basic pay. Other benefits such as Medical and Educational Claims can also be availed in certain companies. All the earnings for a month is together called as Gross Pay. This need not be the same as what salary an employee takes home. This is because on the other side of the weighing scales is the deductions made from the gross earnings.

There are certain standard deductions to be made from the gross compensation payable to an employee. One amongst them is, deduction for absence from duties. In every organization the leave rules determine how many days an employee can stay away from duty and still be paid for. Normally, companies give Casual Leave (CL) upto 12 days in a year. Apart from this there might be Earned leave, also called privileged leave which is granted depending on the length of services of the employees. In case of sickness, a provision for Medical leave might be there. Married women who have worked in the organization for specified length of time are eligible to avail maternity leave at the time of birth of their little ones. Any leave taken over and above the ones mentioned above leads to deduction of a certain amount from the compensation received by the employee and is counted as Leave without pay (LWP) or leave on loss of pay (LLP).

A second major deduction from the gross earnings is on account of Income Tax. Income Tax is deducted for all employees of the private and public sector whose income falls in the taxable category. It is deducted before disbursing the pay to the employees and given to the government. This is known as tax deduction at source.

Another deduction from the employee's gross income is in lieu of Provident Fund (PF). Under the PF scheme, for all employees who have served the organization for atleast 60 days and if there are a minimum of 20 employees in that organization, a certain percentage of the basic pay is deducted from the salary. An equal amount is contributed from the organization and then accumulated in a savings account under the rules and regulations of the Government. This scheme ensures forced savings for the employee.

There is also a provision in most companies whereby employees can take loans against the salary. The loan repayment then appears as a deduction in their payslips. In certain cases where gross earnings of the employee is less than 3,500, contribution from the employee as well as from the employee's side is compulsory. Against this contribution, called **Employee State Insurance** (ESI), several facilities can be availed such as

1. Free medical treatment
2. In case of death of an employee, the ESI pays pension to the surviving spouse
3. In case of accident/long illness where the company does not pay any salary to the employee, ESI compensates him accordingly.

After all deductions have been made from the gross salary, what the employee finally gets is termed as his Net Earnings. In addition to formulation of salaries and wages, the payroll department keeps track of the other kinds of disbursements made to employees. This includes the loan scheme and retirement benefits. Retirement is given to an employee when he attains a certain age preset by the organization (usually 60). The organization compensates all employees who retire after serving the organization in various ways. At the time of retirement the employee takes his compensation as a lump-sum amount, or takes it on a monthly basis.

Pension is the periodical compensation paid to only the regular employees. The amount received every month depends on the number of years of service with the organization. In some organizations pension stops after the death of the employee. The lump-sum payment scheme is called the Superannuation scheme. Under this scheme the employer, the employee and an insurance agent decide on how it can be given to the employee. According to the decision reached, it can either be taken in a lump-sum or a part of it in lump-sum and the rest in installments.

Another compensation paid to an employee in a lump-sum after the employee's retirement or termination of service is Gratuity. every organization which is atleast five years old has to compulsorily pay gratuity to its employees.

The most important aspect of wage formulation is the adequacy of wages, which is divided into two main components : internal adequacy and external adequacy. While fixing compensation it has to be kept in mind whether the amount of money would be sufficient for the employee to maintain a decent standard of living. This is referred to as **Internal Adequacy**. On the other hand, the organization has also to keep in mind what other similar industries are paying to their employees. This is called as the External Adequacy. Various acts are enforced by the Government, the main purpose being to protect workers from being exploited by their employers and to ensure timely and exact cash payments. Most of the workers involve in Trade Union Movements which makes demands regarding minimum compensation levels, bonus which is productivity linked and profit sharing with workers.

"Attendance data" of each employee is the basic information required as an input to the payroll system. It is essential to capture this data accurately. Each organization has its own means for capturing employee attendance data. In some organizations there is a time-office where the time-in and time-out of each employee is recorded. The methods which are followed to capture employee attendance may be Time Tickets and Attendance Registers.

Time Tickets indicate the time spent on a specific job. It consists of employee's name, number, department, regular work hours, overtime hours, etc. It is a means for recording the attendance data of the wage earners / part time / casual workers. An **Attendance Register** consists of names of all the employees department-wise. Each morning the employees are required to sign against their name and indicate the time of arriving into office. In the evening again the signatures and time-out are required. Attendance registers normally record the attendance data of the regular employees.

WHY COMPUTERIZATION?

Since Payroll relates to every employee in the organization, the accuracy and efficiency with which it is prepared can have a critical (and crippling) effect on the employee-employer relations. Moreover, it must be complete in a tight time-schedule. The system produces much information to be sent to external bodies like insurance, provident fund organizations, banks and even the top management.

In the manual payroll, chances of delay in reporting could not be ruled out. But in a computerized system, these fears are totally eliminated. At times, the entire payroll procedure becomes quite complicated because it is governed by statutory regulations, contractual arrangements with employees, and business policies which are continuously changing. In these circumstances again, the computer proves more reliable than humans.

So payroll is generally one of the applications to be computerized at the start in any organization and the results favorably indicate the useful applicability of computers in this sphere.

THE COMPUTERIZED SYSTEM

Broadly speaking, the payroll system maintains detailed records on all employees and permits authorized personnel to obtain a variety of reports on an individual employee or a selected group of employees. The Payroll Data file of the organization contains extensive information on each employee's pay history with the company, pension plan, vacation time accrual, sick time accrual, earnings and deduction for the current month, current quarter and current year-to-date.

Payroll may be calculated on a weekly, biweekly, bimonthly or monthly basis or any combination of these. The system handles hourly and salaried employees and several classifications of overtime and special compensation may be included. Four distinct stages of payroll processing can be identified at any organization.

Employee File Maintenance

The Payroll system has a program which allows adding of new employee details to the employee master file or to change information like name, address, martial status, for employees on a daily basis. Another file, the Monthly Salary Register maintains the employee's monthly earnings and deductions. But the payroll file maintenance procedures do not include provisions to delete employees. The reason is that the records of employees who leave the company are needed at the year end for inclusion on government reports. Therefore when an employee leaves his record is marked inactive but not deleted from the file. Deleting inactive records is done later as a part of year end processing.

Pay Period Processing

This includes several processing tasks for generating employee payslips every month. Pay Period Processing begins with entering and editing of time tickets and attendance register data, which is checked for any obvious errors before the validated information is passed to the payroll processing program. The pay processing program then combines this data with the data from the Employee Master File, performs the necessary calculations for the employee's pay check and updates the employee's yearly wage deductions amount. A Payroll register is printed which is validated manually before printing the payslips. A copy of this register is sent to the financial accounting system.

The final step in pay period processing is to update the employee master file with the hours worked and leave availed information from the time cards and attendance registers. Current gross pay and taxes withheld etc., are also added to the corresponding master file fields.

Quarterly Processing

Quarterly Processing includes several processing tasks for fulfilling required reports that must be sent to the top management and the government agencies at the end of each quarter. For instance, government agencies need a report on the income tax, withheld during the quarter from employee payslips. A Payroll Summary for the quarter may also be taken.

Year-end Processing

It starts with deleting the records of employees no longer working at the company. Before the first pay period, a file maintenance program is run on the appropriate files to enter the new tax rates. And finally, the operations department of the computer centre copies all the disk files to backup tapes, which are stored in the tape library for future reference.

Thus the computerized payroll system elevates the organization to a high level of sophistication.

ORGANIZATION PROFILE

PRICOL which stands for PRemier Instruments and Controls Limited was established in the year 1974 at Perianaickenpalayam, Coimbatore. It is a market leader in the design, development, manufacture and servicing of dash board instruments for the automotive industry. It also manufactures instruments for defence vehicles, switches and its components, oil pumps, digital counters for textile industries, industrial pressure and temperature gauges and machine tools. Currently about 1150 products are manufactured and there are proposals for developing more.

This multi-product organization's annual turnover is about Rs.40 crores. More than 1600 employees are employed currently, both in the factory which is situated at Perianaickenpalayam and at the administrative office which is situated at Coimbatore.

It has a highly computerized environment. It has been continuously replacing and upgrading its hardware and software to keep abreast with the new technology. PRICOL currently has one Magnum and a HP/9000 series 817 machine with 70 terminals at the factory and one Magnum and one HP/9000 series 832 machine with 40 terminals at the city office. Data transfer among machines at factory, city office and ancillary units is through tapes, DATS, and floppies on a daily basis. For CAD/CAM work PRICOL has one Apollo Workstation with SDRC software and three Omega Workstations and several PC-ATs using AutoCAD Software.

Major part of the application softwares are developed in COBOL and some in FOCUS. Recently they have started developing new systems and converting existing systems to SYBASE to keep up with the new technologies. Payroll System is one such application.

The success of this organization lies primarily with its new and innovative ventures in the field of electronics.

PROBLEM DESCRIPTION

PRICOL employees are categorized as Regular Staffs, Operators, Field Staffs and Shop Floor Operators. Each employee comes under one particular category and grade which is determined by the top management according to the nature of their job, educational qualification, experience, etc. Each category and grade has its own pay structure which consists of Basic salary, increment amount, HRA, Vehicle allowance, CCA, per shift allowance, eligible leave days, bonus amount, union subscription, etc.

The Payroll System is used to calculate the employee's salary for the month. Every month, payroll for the previous month is run before 3rd of the current month and the salary is distributed on or before 10th of the current month. Payroll for the field staffs is executed by the 28th of the month itself, since they are stationed outside Coimbatore.

Earnings consist of earned Basic, HRA, fixed DA, variable DA, CCA, Vehicle allowance, Attendance bonus, Shift allowance, etc. Variable DA and attendance bonus are only for operators. After earnings calculation fixed or compulsory deductions like EPF, FPF and ESI are done unconditionally. A part of the gross salary is preserved as take home salary for the employee before any other deductions start. The percentage of take home salary depends on the employees category and grade. Other deductions are done only on the left over salary.

Other deductions like canteen amount, co-operative stores amount, Loans, Advances, etc are deducted from the left over salary in order of the priority assigned to each type of deductions. The deductions are done partially or fully depending on the partial/full nature of each deduction. After the loan type deductions, the number of balance installments and non deducted amount for the month are maintained and other type deduction information are kept in deduction transaction table.

They have a PRICOL FINANCE division. Employees take loans and also make investments at the Finance company. These particulars are maintained in the Payroll System also. There is also a Thrift Society for the benefit of the employees. Members of the society can buy home appliances at lower price and pay back in easy installments at lesser rate of interests. There is a bus which runs from the city to the factory. Bus amount is deducted from the employees who have a Bus Membership. Subscription amount for the various labour unions are also deducted. LIC and Post Office deductions are made for the respective policy holders. Several other such deductions have to be made. No matter how much deductions have to be made the take home salary is given to the employee and the non-deducted amount is accumulated and carried over to the forthcoming months. The payslip is printed on pre-printed stationery and several reports are taken for the various levels of management.

DRAWBACKS IN THE EXISTING SYSTEM

One major drawback in the existing system is that in case any mistake has been made in the attendance entry there is no facility to modify the details. This is because the attendance details are entered while processing pay calculations into the pay transaction table. Only after the calculations are completed the error is made known to the operator. Moreover attendance details are stored along with pay calculation information in the pay transaction table. If the wrong attendance information is to be rectified, the pay processing has to be run once again. If the pay processing has to be run again the pay transaction records created have to be deleted which results in loss of attendance information. The attendance for all the employees have to be re-entered. To avoid entering data once again they take frequent backups at each stage of .pathe processing. However this process is very tedious and time consuming.

GENESIS is a primitive RDBMS. It is not capable of handling the vast database efficiently. The system performance is poor. Even to switch between menus it takes at least a minute. It sometimes takes hours to process payslips and even more to print them. Also it does'nt display the processing time. One never knows whether the system is processing or idle. It takes about 3 days for the operators to complete the pay processing job for all the employees. Because of the slow speed they sometimes split the processing into batches of employees ie., they run the processing for a particular category at a time.

Moreover, GENESIS is now becoming an out of date software. There are no software professionals readily available to make modifications in it.

A lot of reports are taken at the end of the month for the MIS department and the top level management. The reports are taken in various combinations - department wise, category wise, grade wise, and so on. Each and every report is generated and stored in a separate spool file. The spool files are then directed one after the other to the printer manually. It is also difficult to keep track of each spool file name and which of them are yet to be printed. A menu providing the user with all the options was suggested. The user can just select all the combinations of reports necessary at a stretch.

There were other internal calculation problems. For example, for a newly joined employee at the middle of the month the number of payable days was calculated incorrectly.

Another serious problem in the existing system is that the non-deducted deduction amount that is accumulated is not carried forward to the next month's deduction. The accumulation is manually re-initialized at the end of the year.

Some problems in the design of the database were also identified. For each category of loan, savings scheme and advance a unique code is assigned. But there is no provision for an employee to avail a particular loan (or any other deduction type twice). For example, a person could not have more than one LIC policy.

The number of extra shifts worked by an employee could be entered only as a whole number since the field was declared to be of integer type.

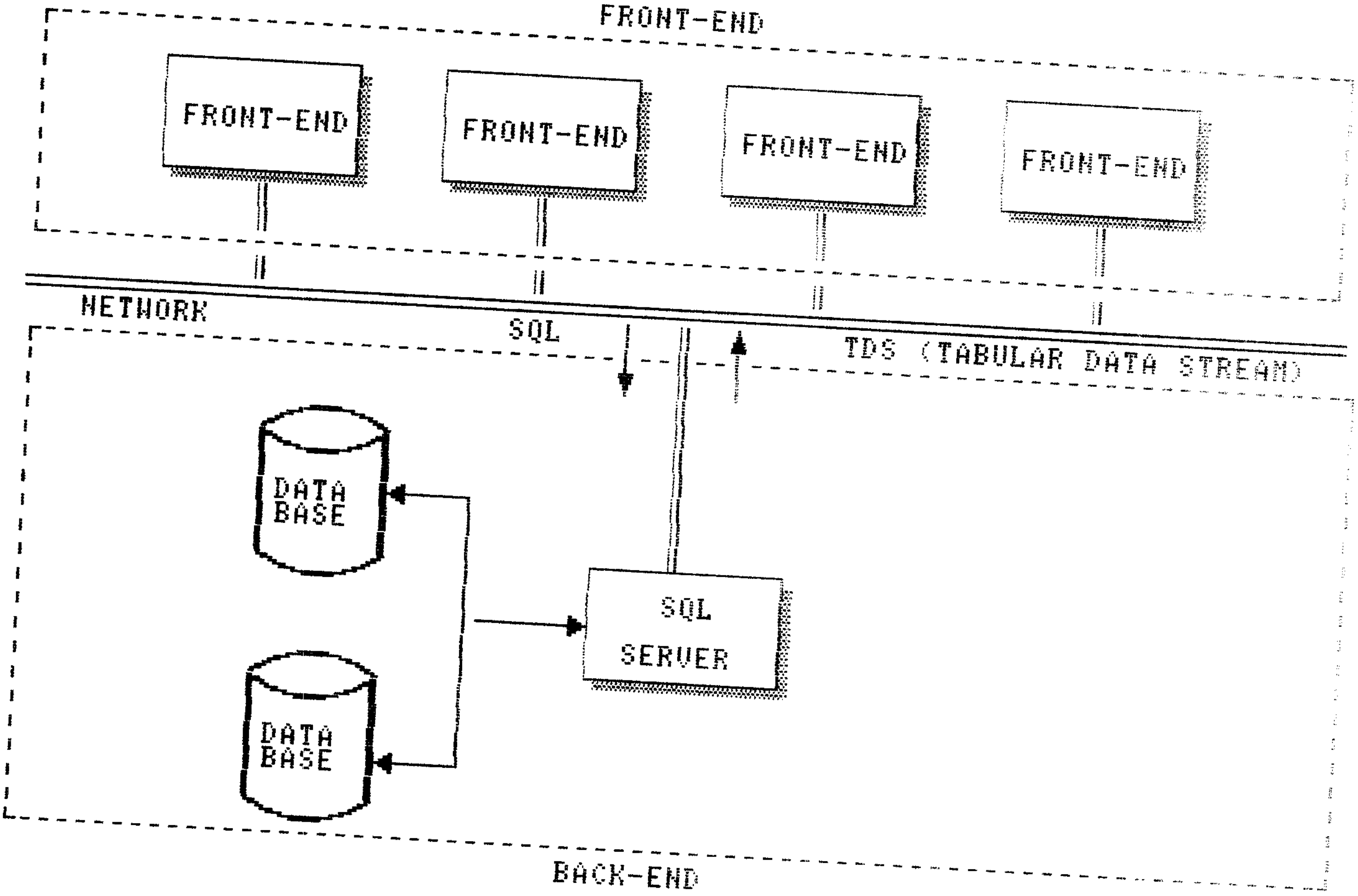
Also lot of unnecessary temporary tables and history tables were created which led to data inconsistencies. Thus a new system proposal was made to overcome the existing drawbacks.

SYBASE AND CLIENT/SERVER COMPUTING

INTRODUCTION

The Client/Server computing model covers a wide range of functions, services and other aspects of the distributed environment. The Client/Server computing model implies a co-operative processing of requests submitted by a client or requester to the server, which processes the requests and returns the results to the client.

SYBASE handles on-line applications with its advanced Client/Server Architecture. With this architecture, data management and transaction processing functions can be performed independently from client applications and user interfaces. The front-end (client) and the back-end (server) are connected by a network. The front-end deals with the user interface and the back-end executes SQL statements and deals with database management. The data is transmitted through the network in a tabular form called the TDS (Tabular Data Stream). The SQL server, as the back-end, manages multiple users and multiple tasks, thereby reducing the operating system overhead and improving performance.



CLIENT/SERVER ARCHITECTURE

SYBASE clients and servers can be deployed on the same machine or on different machines, communicating transparently across a network of microcomputers, minicomputers and mainframes. This is exactly what a Distributed Database Management System (DBMS) does.

CLIENTS ROLE AND FUNCTIONS

Since client nodes are typically designed to interact with end users, their functionality and implementation can be specialized for these interactions. Similar to developing specialized sensory cells in humans and animals, client nodes must be designed to deliver such functions as high-resolution graphics and sound at acceptable performance levels. However, complete application requirements often exceed those of client node interactions and can be satisfied only as a result of the collective performance of all system nodes.

The major functions performed by a client system in a Client/Server environment are presentation functions and some business logic. End user interactions with an application are performed through the presentation logic. The presentation logic is the layer of the architecture that, on one hand interacts with the business logic of the application, and on the other hand, interacts with end users. The latter includes all interactions with the physical device (terminal) and handling of actual end-user and performed input/output (screen I/O, keyboard I/O, mouse, etc.).

Traditional presentation functions dealt with character-based displays, where the processor displayed characters received from an application sequentially in a fixed font on a screen. Continuous evolution of presentation functions has been closely linked with high-performance workstations offering graphical display capabilities. These displays allow the processor to control individual picture elements (pixels) on a screen. Modern client workstations can combine these display capabilities with images, full-motion video, sound, optical character recognition, pen-based input and interactive multimedia. These capabilities allow software designers to create uniquely effective, intuitive application interfaces that can actually blur boundaries between applications.

SERVER FUNCTIONS

Architecturally, a server is a logical process that provides services to requesting processes. In Client/Server computing, a client initiates the Client/Server interaction by sending a request to its server. The functions that a server should perform are determined, in large part, by the types of requests that clients can send to the servers. Conversely, if a server is unable to perform a function requested by a client, then this server cannot participate in co-operative Client/Server interactions. Ideally, a client should not be sending an unsupported request to such a server. In general, however once clients and servers are interconnected in a network, the following functions may be required of servers by the users:

File Sharing

In a workgroup environment, clients may need to share the same data file, for example, an insurance rates file in an insurance office. The rates file is placed in a shared file processor - a file server - and clients send their file I/O requests to the file server. Usually, a file server provides a client with the access to the entire file, so that when one client updates a shared file, all other clients are unable to access this file. Another typical use of file servers is for a file transfer between clients.

Printer Sharing

In a workgroup environment, one high-capacity printer may replace all individual client printers. Then all clients may send file print requests to a print server. A print server maintains a queue of all files to be printed, sending each print file, in turn, to a shared printer (usually, a high-output, high-quality printer). Typically, all individual print files are printed with a special separator page that indicates the client name and file name.

Database Access

In a Client/Server environment, application processing is divided between client and server systems. Servers may execute some portion of the business logic and database logic. Similar to file servers, database servers provide clients with access to data that resides on a server. However, database management systems are more sophisticated than basic file I/O access methods. DBMSs provide concurrent data access with various levels of locking granularity and data integrity. DBMSs eliminate data redundancy, allow for user transparent data distribution, and even allow parts of application- specific data access logic to be incorporated into the DBMS itself. Clients request access to desired data (contrary to a file server's access to the entire file), and all necessary manipulation on the required data is performed at the database server. Thus, multiple clients can access a database concurrently.

Communication Server

In a workgroup environment that is connected to a remote host processor, all communications software and hardware can be concentrated on a special communication sever to which clients may forward their communication requests for processing.

Fax Servers

FacSimile services which usually require special equipment and software, are now more frequently trusted to dedicated fax servers. Clients send and receive fax documents by requesting appropriate services from a fax server.

Other Client/Server functions, such as electronic mail, library, network, resource, and configuration management, are being handled in today's Client/Server environment by appropriate severs. A server node in a Client/Server model can be specialized to perform its particular function in the most efficient way. However, besides individual, function-specific specialization, servers as a class of systems can be specialized to satisfy the following general-purpose requirements.

Multiuser Support

Even in a small workgroup environment, a server should be able to service multiple concurrent clients. Clients running different tasks would expect a server to support multitask processing. Note that a multitasking system is not equal to a multiuser system.

Scalability

Scalability is the property of a system that permits an incremented increase in capacity, performance, throughput, the number of supported users, and so on, by adding required computing resources as needed and without changing the applications. As the number of applications, their resource requirements, and the number of users grow, a server should be able to satisfy these increasing demands on its resources; ie., it should provide scalable performance. Scalability does not mean that users should buy an overcapacity server system at extra cost. On the contrary, the system should satisfy current requirements and, at the same time, should be easy to expand. This expansion can be achieved by vertical scaling of a server or by horizontal scaling in which multiple users co-operate transparently to share the workload.

Performance and Throughput

A server system should provide performance and throughput levels satisfactory to the business needs and user requirement in a multiuser Client/Server environment. For example, even if business requirements do not call for subsecond response time for every business transaction. users would hardly appreciate a system that takes more than a few seconds to respond to every user action. Similarly, if the workload on a server increases with the addition of new servers, neither the performance nor the throughput should suffer. A server system should be able to provide scalable and easily tunable performance and throughput.

Storage capacity

As the number of users and applications running on a server increases, and as advances in storage technology drive the costs of physical storage down, the demand for extra storage and faster access times becomes one of the critical requirements for a server system. The storage demands come from operating systems that needs additional storage to implement new advanced features, from users who wish to store various data files on a server, and from applications such as DBMS and CASE tools that are some of the major storage consumers.

Availability

As more and more mission-critical applications are migrated or deployed into a Client/Server environment, the availability of the server system becomes an essential business requirement. Similar to the mainframe data center environment, today's servers are expected to be up and running most of the time, and the 24 x 7 (24 hours, seven days a week) uptime for a server is not at all unusual. The key factors affecting availability are the server robustness and online administration. Robustness implies that the server system reduces the importance of any particular failure and recovers from it transparently and automatically. Hardware and software fault tolerance, including features like hot and warm standby servers, disk duplexing and mirroring, and the use of RAID (Random Array of Inexpensive Disks) disk subsystems, are all designed to improve robustness. Cperations such as database recognition, backup and recovery, starting and stopping of server processes, system monitoring and configuration user administration, and application/system upgrades should ideally be performed online, without taking the server down.

Multimedia

As new applications and new technologies become available, the demand for multimedia storage support is increasing. Image, video and sound applications are becoming more and more popular. So, the requirements for a server system may include the ability to store, not only digitized images on disk, but also hypertext on an optical storage device - a WORM (write-once-read-many) - and video/sound data on video cassettes, compact discs and video disks.

Networking and Communications

Client/Server communications happen over a communication network. Both client and server systems have built-in networking capabilities. Without networking there is no Client/Server interaction, therefore, no clients and servers. Hence the system hardware and software architectures should be optimally integrated with the networking interfaces and protocols.

SYBASE SQL SERVER ARCHITECTURE

One of the reasons for the success of SYBASE is its unique multithreaded architecture. A unit of context management under the control of a single process is called a thread. A thread can be either implemented within the server process or via operating system services. A considerably more efficient approach is to launch a separate thread for each separate task. Such a "lightweight" task can be controlled by the DBMS server rather than by the operating system.

Among UNIX - based RDBMSs server implementation, SYBASE SQL Server is one example of the multithreaded server architecture. SYBASE's latest release of SQL Server extends the concept of the multithreaded architecture to symmetric multiprocessing hardware platforms. The multithreaded database server can manage all the resources needed by the RDBMS itself (including buffers, disk I/O, locking and logging) which essentially makes it a special-purpose operating system dedicated to DBMS operations by scheduling threads execution.

The SQL Server engine uses asynchronous device I/O. It allows the user to partition a physical device into several logical devices. Within the dataserver process, each logical database device uses a separate thread, providing asynchronous I/O capability even within one physical device. Because a separate thread is used for each logical device, the SQL Server can write concurrently to different database devices. The asynchronous I/O capability also allows for nonserial writes within mirrored devices. Without asynchronous I/O, the system would wait for each write to finish before starting another.

Client applications can benefit greatly from the use of asynchronous programming capabilities. Asynchronous applications are designed to make efficient use of time by performing other work while waiting for server operations to complete.

Multiprocessor Servers

The SYBASE SQL Server design is based on SYBASE virtual server Architecture (VSA) which allows it to utilize parallel processing features on symmetric multiprocessing (SMP) Systems. SQL Server can be run as a single process (does not utilize SMP) or as multiple co-operating processes. Depending on the number of CPUs available and the demands placed on the server, a good practice is to run a maximum of one SQL Server engine .paper available CPU. The SYBASE definitions for its specific features are as follows.

Process

An execution environment scheduled onto physical CPUs by the operating system.

Engine

A process running a SQL Server that communicates with the other SQL Server processes via shared memory. An engine can be thought of as one CPU's worth of processing power. It does not represent a particular CPU.

Task

An execution environment within the SQL Server scheduled onto engines by the SQL Server.

Affinity

Describes a process in which a certain SQL Server task runs only on a certain engine, or that a certain engine runs only on a certain CPU.

SYBASE FEATURE SET

SYBASE SQL Server has earned a reputation for reliability and high performance, making it the relational database management system of choice for online transaction processing (OLTP) applications. Some particular strengths of SYBASE SQL Server are :

* 32-bit multithreaded architecture
* Dynamic (online) backup/continuous operations
* Database triggers and stored procedures
* Scalability and flexibility
* Enforcement of business rules and referential integrity
* Support for user-defined data types
* Third-party front-end support
* Product maturity/breadth of product line

ADVANTAGES OF CLIENT/SERVER COMPUTING

The real benefits to adopting a Client/Server Architecture are as follows :

* Allows corporations to better leverage emerging desktop computing technology. Today's workstations deliver considerable computing power, previously available only from mainframes, at a fraction of mainframe costs.
* Allows the processing to reside close to the source of data being processed; therefore network traffic can be greatly reduced, and effective throughput and carrying capacity on a heavily loaded network is increased. Conversely, the network bandwidth requirements and therefore, the cost can be reduced.
* Facilitates the use of graphical user interfaces

(GUI) and multimedia applications, available on powerful workstations. These new interfaces can be delivered to customers using a variety of visual presentation techniques, together with easy and intuitive navigation and standards - compliant consistency. Indeed, a picture is worth a thousand words. As a result, investment in training and education can be leveraged better and new products that exceed customer expectations can be developed faster. End user resistance to accepting new products can be minimized.

* Allows for and encourages the acceptance of open systems. Indeed, the fact that clients and servers can, in fact, be running on different hardware and software platforms allows end users to free themselves from particular proprietary architectures, thus taking economic, marketing and competitive advantage of the open market of available products.

THE SYBASE RDBMS

Database objects are the main components of a database in which the Server stores all related data pertaining to it. The database objects in SYBASE are :

Tables

Tables are the primary objects of a database since they contain data in the form of columns and rows. The tables are created after the database is created. The data values in a table must be normalized.

Views

Views are virtual tables consisting of a subset of the rows and columns in one or more tables. In addition to providing a level of isolation from restructuring in the database and changes in naming conventions, views can be used to enhance security by restricting the visible data from the underlying tables. A user can be granted permissions on a view, even if he or she has no permissions on the tables that the view references.

Views can be used to hide data from a user or group of users. By accessing data using a view, a user can query and modify only the data that is visible through the view. The rest of the data is not accessible to the user. Views can be used to restrict access to both column subsets and row subsets of a base table. Views allow users to access only that subset of data required to perform their job function. Views can also be used to restrict access to a subset of another view, or of some combination of views and tables.

Data types

Datatypes define the kind of data that can be stored in a column or a variable. They may also be user-defined. Depending on the size of the numeric data, int, smallint or tinyint is chosen. Float is used to specify floating point type data. Money is used for dollar and cent fields. Varchar is used in place of char for fields which vary a lot in length, such as names. Datetime fields hold the date and time. Text types are used to hold large text items in the range of 2K. User-defined datatypes can be used wherever system datatypes cannot be used. They provide consistency among columns that are repeated in different tables due to their ability to bind rules and defaults.

Stored Procedures

Stored Procedures are precompiled SQL statements that are stored on the SQL database server. Stored Procedures can accept command-line arguments and can also return values to the calling program. Stored procedures offer several benefits :

* The SQL code on the server is already optimized and compiled.
* Reduced network traffic. A client can send two words across the network (ie., "execute procedure_name") rather than sending the hundreds of words that may be in the procedure.
* They provide a single point of code maintenance (at the server, not the client workstations or application programs. In a typical UNIX-based Client/Server environment, for example, a stored procedure can be processed in one-fifth the time it takes to process a single embedded SQL statement. Frequently used queries are likely candidates for stored procedures.

Triggers

Triggers execute in response to a database event. Triggers are similar to stored procedures and are associated with a particular table. They are automatically invoked by the server in response to a database event (whenever the table is the target of an INSERT, UPDATE or DELETE operation). Triggers cannot be executed manually.

The SQL statements that define the trigger are executed when the triggering statement (the INSERT, UPDATE or DELETE) is completed. In the event of an error, the trigger can cause all actions performed by the trigger and the triggering statement to be rolled back. Only one trigger can be defined for each operation on a particular table. A single trigger can be defined for all operations associated with a table. Triggers can be quite complex and as a result, they are more flexible than the declarative integrity constraints.

Triggers can be used to compare the before and after images of a row or rows during a transaction. Within the batch, SYBASE uses two virtual tables called the inserted and deleted tables. These tables are local to the trigger, that is, a trigger can cause other triggers to fire and that trigger will also have access to inserted and deleted tables. The inserted table contains copies of rows that have been inserted into the target table as a result of an INSERT or DELETE operation. The deleted table contains copies of rows that have been removed from the target table by the effects of a DELETE or UPDATE statement. UPDATE will cause a row to be placed into both tables(inserted and deleted). In this case, the deleted table contains the before image of the row and the inserted table the after image.

Rules and Defaults

Rules and Defaults are provided to help maintain entity integrity and domain integrity. Entity integrity is used to ensure that a value is entered for all columns that require a value; domain integrity makes sure that each value in a column belongs to a set of legal values for that column. Defaults and Rules define integrity constraints during the entry and modification of data.

A Default is a value linked to a column or data type that is inserted when no value is provided during data entry. Rules are user-defined integrity constraints also linked to a column or data type and enforced at data entry time.

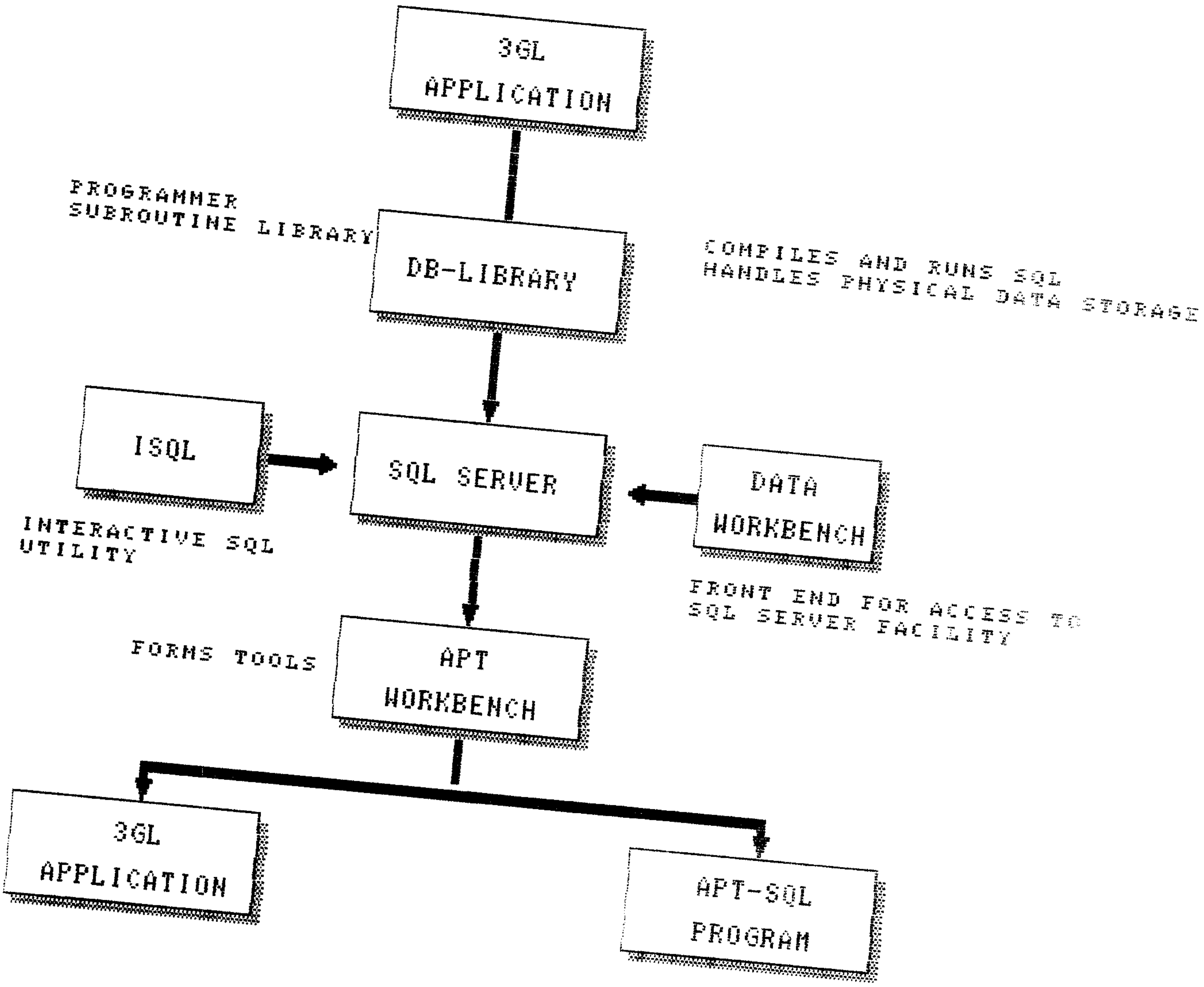
COMPONENTS OF SYBASE

SYBASE provides various front-end tools for developing on-line applications. Some of them are as follows:

ISQL (Interactive Structured Query Language)

ISQL is an Interactive and batch SQL interface to the SQL Server. In interactive mode, ISQL can be invoked from the command line by typing ISQL with optional command line arguments. ISQL can be used to produce simple tabular and summary reports. All SQL Server operations can be performed using the ISQL utility. ISQL is not an end-user tool and is intended for use by system administrators and programmers. It is particularly useful for running batch jobs, database and query tuning, capturing output, and connecting to Open Server applications.

COMPONENTS OF SYBASE



APT Workbench

APT (Application Productivity Tool) - Workbench is a forms-based 4GL development environment created and marketed by SYBASE. It is an environment for developing character-based applications that can operate in the heterogeneous computing environment typical of Client/Server systems. APT Workbench is a collection of tools for developing forms-based menus and processing. The Application Productivity Tool includes APT-Edit™, APT-SQL™, APT-Build™ and APT-Execute™. APT-Build is an automatic application building tool, and APT-Execute runs applications that are under development and provides an end-user environment for running completed, compiled applications. APT-Library™ is a set of forms processing routines that can be called from a host language program. APT-Edit is the forms editor of APT Workbench. It provides a point-and-pick method for developing forms and form processing for applications. One has to just pick menu options and fill out property sheets as required for your applications. The form is stored on the host machine and remains completely independent of the database.

DWB Data Workbench

Data Workbench is an interactive data access and administration tool. It can be used for administration and data entry and to generate quick reports. It is a collection of front-end tools which provides facilities to format query results, generate reports using the Report Writer and to do basic data entry in a faster and easier manner.

DB-Library

DB-Library is a call-level interface used to write client applications. DB-Library includes a bulk copy library and a special two-phase commit library. DB-Library was the only library available in earlier versions of Open Client and is carried over from earlier SYBASE versions. DB-Library is a library of calls that can be called from a host language program to interact with SQL Server. APT-Library and DB-Library are part of any real-time application including forms and SQL Server. It has a set of C routines and macros that allow an application to interact with the SQL Server, ie., to transmit SQL statements to the SQL Server and return data to the application program. While the requests to the SQL

Server are in SQL text, the results passed back to the application are in host language datatypes. Versions of DB-Library exist for a number of languages besides C, including COBOL, FORTRAN and PASCAL. The host language used in this project is C. An application protocol called 'TDS' (Tabular Data Stream) is used by the SQL Server to return results ie., the data returned is in rows and columns. The DB-Library approach is distinctly different from the "embedded SQL" type of host language interface that some DBMSs provide. In contrast to an embedded SQL interface, the DB-Library interface does not require a host language pre-compiler. Avoiding preprocessing in this way makes applications involving a database more straight forward to write and debug. The main functions of DB-Library are :

- Manage communication and data transfer between SQL Server and the application
- Provide a consistent programming interface across multiple languages
- Provide subroutine / function calls
- Provide standard structure and type definitions
- Insulate application programs from the changes to SQL Server internals.

REFERENTIAL INTEGRITY

For example, a stock broker maintains positions for an account by executing trades on behalf of a client. In this scenario, the account table is related to the position table. Since there can be multiple positions for an account, the relationship is said to be one-to-many. When these tables are joined, the account table is the parent and the position table is the child. If a parent row is deleted and the related child rows are not, the child rows are said to be orphaned. Referential Integrity means that no orphans are permitted in any table. File/Server databases expect the programmer to include logic to enforce Referential Integrity. Most Client/Server SQL databases include RI features.

SYBASE provides support for the two types of Referential Integrity.

Declarative Referential Integrity

Declarative Referential Integrity is implemented through keys that are stored within the database tables themselves. Parent tables contain primary keys that are composed of foreign keys found within each one of the child tables.

Procedural Referential Integrity

Procedural Referential Integrity is implemented through the use of triggers. Triggers ensure RI by automatically executing SQL statements that enforce business rules whenever an INSERT, UPDATE or DELETE command is performed on a table. For example, whenever a parent row is deleted, either all of the child rows are deleted (cascading deletes) or an error condition is reported and the deletion is disallowed.

DATA INTEGRITY

Database transaction can be defined as a sequence of one or more data manipulation statements that together form an atomic, logical unit of work. Either all statements in the transaction will execute successfully or none of the statements will be executed. A transaction is said to be committed when it completes all processing successfully. When the request for data modification (transaction processing) is committed, it is written to disk from cache memory. SQL databases maintain a transaction log that records information about all modifications that have been requested to a database. SYBASE SQL Server uses transaction logs for transaction rollback (the ability to reverse committed transactions and restore a database to its previous state).

The Two-Phase Commit (2PC) logic ensures data integrity when updates are performed in a distributed database environment. Two-Phase commit forces a database update to be a two - step process

- The preparation to commit process where each Server says "I'm ready"
- The actual commit process

SYBASE provides a programmable Two-Phase Commit while some other RDBMSs supply automatic 2PC. SYBASE Open Client DB-Library provides a special Two-Phase Commit library for application programs. Transaction rollback is used in user-defined transactions with the BEGIN TRAN, COMMIT TRAN, and ROLLBACK TRAN commands.

SYSTEM DESCRIPTION

The Payroll System can be broadly classified into the following 5 modules, namely, Daily Jobs, Monthly Jobs, Yearly Jobs, Master Maintenance and Adhoc Jobs. These 5 tasks form the items for the main menu. The description of the tasks and their sub modules follows in the order of categorization.

DAILY JOBS

The day-to-day entry jobs are done using this menu. All claims made by employees such as Medical Claim, Leave Travel Allowance Claim and Education Claim are entered in this module. Details such as the Employee Code, the Bill number, date of claiming and the amount claimed are entered into a table called MELDET. If the employee's allowance accumulation account has enough amount to meet the claim, then the claim is accepted else rejected. As an exception, if the employee has obtained the permission from the MD then the claim is accepted even if his/her allowance accumulation account is not sufficient to meet the claim.

After accepting a valid employee code and allowance type, the current date and bill number are displayed from the CONTROL table and the claim amount is accepted and validated. On confirmation from the user, the loan details entered for an employee is updated in the MELCONS table and the bill number is incremented in the CONTROL table.

At present no other process is involved in this module and no reports are also taken.

MONTHLY JOBS

The monthly transactions are invoked using this monthly jobs menu. The activities under this item are classified as

- Data entry
- Processing Jobs
- Reports

Data Entry

Monthly data entry for the pay processing is done using this menu. The items of this menu are

- Attendance details
- Variable earnings
- Variable deductions maintenance

Attendance Details

This is an important item in the pay processing. It is through this screen the attendance details can be entered for a particular employee or for a set of employees. The attendance details can be created for all the employees for the particular month with the default attendance. These records can then be altered if necessary. The maximum number of working days is got from the CALENDER table. The number of Casual Leave (CL) Days availed by the employee is entered. If the employee has applied for more than the CL days available for him then the balance days are treated as Leave on Loss of Pay days (LLP).

If an employee has ESI and takes medical leave, then the salary for the absented days will be paid by the government. The absented days are known as ESI days. If an employee was absent without prior intimation to the company then these days are treated as absented days. If the employee's leave is not sanctioned by the company then these days are treated as not sanctioned days. If any compensation has been made for the leave due to strike etc., the number of compensation days is entered. If an employee is late to the factory/office the total number of minutes/hours by which he was late is entered from the time card. The number of days worked during the month is also entered. Other details such as layoff days are also entered for operators. The details are entered in the table called ATTN.

Variable Earnings

This module allows the user to enter the variable earnings for the particular month. The number of extra shifts worked by an employee constitute the basis for the variable earnings. After entering a valid employee code and the number of extra shifts worked by the employee for the month is calculated as,

number of extra shifts worked * per shift amount.

The valid employee code and the number of extra shifts along with the month and year are entered in the table VAREARN.

Variable Deductions

Deductions are classified into 3 types, such as, loan type deductions, non-loan type deductions and deductions that are made constantly every month. According to the type of deduction availed by an employee, the deduction entries are made in the respective table. For example, deductions like Personnel loan, Education loan, Dhanalakshmi Loan are loan type deductions. Deductions like canteen, co-operative stores, Bus amount, etc., are non-loan type deductions. LIC and Post Office RD are deductions that are to be made always.

In case of loan-type deductions the entry is made into a table called INSDED. It contains the employee code, the major deduction code, deduction sub code, the loan number, the date of commencement of the deduction, the principal amount, total number of installments, the rate of interest, the first installment amount, the regular installment amount, the amount deducted so far and the number of installments deducted. There is a flag which indicates the record is active and the deduction is processed. In case the installments are over or the loan is not recovered the flag is set inactive.

Non-loan type deductions are entered in a table called NINSDED. Details such as the employee code, the major deduction code, the deduction sub code, the date and the amount to be deducted are entered. If the flag is active the deduction is processed and the flag is set inactive.

LIC and Post Office deduction entries are entered in a table called LICPODET. It contains along with the employee code and deduction codes, the policy number, the date, the amount and the accumulation amount in case of post office savings.

Pay Processing

The Pay Processing Program is run monthly in batches for each category. The year and month of processing, a valid category and the user's confirmation are accepted as input. The Pay Processing program takes input from various tables for computations.

The possible working days for an employee are calculated as follows.

Possible working days = Maximum days in the month - number of weekly off days in the month

Schedule days = possible working days
- number of national holiday(s)
- number of festival holiday(s)

For the staff there is no difference in schedule days. It is always 26. The maximum days in the month, the number of national and festival holidays for the month are obtained from the

CALENDER table. The number of weekly off days for the employee during the month is obtained from the WEEKDET table. The number of payable days for an employee is calculated as follows.

If the availed CL days are greater than CL days available in the Employee Master then the CL days for the month is calculated as

CL days = CL days available in Employee Master

LLP days = CL days availed - CL days available in employee master

If the availed CL days are lesser than the CL days available in Employee Master then CL days for the month is calculated as

CL days = CL days available in employee master - CL days applied for the month

LLP days = 0

Total leave days, physically worked days and payable days are calculated as follows :

Total leave days = CL days + ESI days + Absented Days + Not Sanctioned Days + Medical Leave + LLP Days

Physically worked days = schedule days - Total leave days - Layoff days

Payable days = physically worked days + CL days + number of national holiday(s) + number of festival holiday(s)

All types of leave days and worked days are retrieved from the attendance table ATTN.

If the employee is a staff, then the physically worked days and payable days are calculated as follows.

Physically worked days = 26 - Total absented days -
No. of national holiday(s) -
No. of festival holiday(s)

Payable days = 26 - Total absented days + CL days -
No. of national holiday(s) -
No. of festival holiday(s)

If an operator has taken leave for less than or equal to one day then he/she is eligible for attendance bonus. The attendance bonus will be calculated as,

Attendance bonus per day = Attendance bonus from employee master / 26

Attendance bonus = Attendance bonus per day * payable days

If an employee is a staff then he can avail fixed DA. Operators are eligible for variable DA. Fixed DA is a lump-sum amount. Variable DA amount is calculated as follows:

Variable DA = (current month index point - fixed index point) * per point amount

Variable DA amount = variable DA / no-of-days * payable days

Basic earned = basic / no-of-days * payable days

CCA = CCA / no-of-days * payable days

HRA = HRA / no-of-days * payable days

Vehicle allowance = CA / no-of-days * payable days

If the employee is a staff then number-of-days = 26. CCA, HRA and CA rates are available in the pay structure table PAYSTRU. Basic is available in the employee master.

Only operators are eligible for layoff amount. If the layoff days is greater than zero, then the

Layoff amount = (((Basic + variable DA + Fixed DA + HRA) / no-of-days) / 2) * no. of layoff days

Shift allowance = No. of shifts * amount per shift

The number of shifts is obtained from the variable earnings table VAREARN.

The gross earnings of an employee is calculated as follows:

Gross earnings = Basic + Fixed DA + Variable DA + CCA + HRA + Vehicle allowance + Attendance bonus + shift allowance + layoff amount + variable earnings

The Gross earnings is treated as the ESI salary. The PF salary is calculated as

PF salary = basic + variable DA + fixed DA

If the employee is a staff else if an employee is an operator,

PF salary = (((Basic + variable DA + Fixed DA + Layoff amount + HRA) / no-of-days) / 2) * Layoff amount

The EPF and FPF amount are calculated as follows :

EPF amount = PF salary * EPF Percentage / 100

FPF amount = PF salary * FPF Percentage / 100

If the FPF amount exceeds the FPF maximum amount then that exceeded amount should be added with EPF amount. EPF and FPF amounts are rounded off to the nearest given rounding off factor which is taken from the CONTROL table.

The ESI deduction is made only if the ESI indicator of the employee is set to Y. The ESI flag of an employee is set to Y under 3 circumstances. They are as follows.

* If the employee is a newly joined one and if the salary is the first for him/her after joining the company

* If the current month is the ESI calculation

month (this is got from the CONTROL table)

* If the month is the second ESI calculation month ie., six months after the first ESI calculation month.

ESI amount = ESI salary * ESI percentage / 100

This amount is rounded off to the highest rounding off factor given in the CONTROL table.

The compulsory deductions are done as given below

Net amount = Gross earned - EPF amount - FPF amount - ESI amount

A percentage of the Net amount is reserved as Take Home Salary for the employee. The percentage differs for staff and operators and is available in the CONTROL table. All other deductions are made from the remaining portion of the salary according to the priority and the Partial/Full indicator.

Late hours deductions are made if the employee arrives at the office.

Late Hours amount = (((Basic + Variable DA + Fixed DA + HRA) / 26) / 8) * No.of late hours

If the deduction is Professional tax, then the slab rate of the employee's professional tax salary is fetched from the PTDET table.

If the employee's salary is not sufficient to deduct the Partial/full indicator is checked. If the indicator is partial, the maximum partial amount is deducted and the rest is included as non deducted amount. If the indicator is full, then the whole amount is included as non deducted amount. This non deducted amount will be carried over to the next month's deductions. The deduction details like the year, month, employee code, deduction major code, deduction sub code, deducted amount and non deducted amount are entered in the DEDTRN table for every deduction made.

The net pay is calculated as,

Net pay = Net amount + Take Home Salary Preserved

Net pay round off is done to the lowest amount for the given rounding off factor in the CONTROL table. If the old paise balance of the EMPMST table and the current month paise balance is greater than the rounding off factor, then the rounding off factor is added to the netpay as a type of earning as,

Net pay = Net pay after rounded off + Rounding off factor

The earnings and deduction values are then inserted into the PAYTRN table. After the pay processing the updation operations are performed. The paise balance in the employee master is updated. The balance Credit CL is also updated. The entries in loan-type deduction table are also updated. The deducted amount and the total number of deducted installments are updated.

New deduction balance amount = Old deduction balance amount - Amount deducted from the salary

New non-deducted amount = Old non-deducted amount + current month non-deducted amount

New deducted installments = Old balance installments+ 1

The CL balance for an operator is updated in the EMPMST table as follows :

Balance CL Days = CL days in EMPMST table - CL taken in the current month

The CL balance for the staff is updated in the EMPMST table as follows :

Balance CL Days = CL days in EMPMST table - CL taken in the current month + Number of festival/national holidays that come on Sundays

The paise balance is calculated as follows :

Accumulated paise balance = Old paise balance + Current month paise balance

If the accumulated paise is greater than the rounding off factor of CONTROL table then the excess amount can be calculated as,

Excess amount = Accumulated paise balance - Rounding off factor of CONTROL table

The paise balance is updated in the EMPMST table as follows :

Paise balance = Old paise balance + excess amount

Reports

This module is used for printing the various reports in connection with the monthly pay processing. The various reports taken are as follows:

* Payslip Printing
* Payslip enclosure
* Salary Distribution list
* Deduction details for the month (dept. wise)
* Earnings details for the month (dept. wise)
* EPF, FPF deduction details for the month
* ESI deduction details report
* Payroll abstract
* Gradewise strength report
* Departmentwise strength report
* Absenteeism report
* Loanwise deduction/ Non deduction report
* Cost centrewise earnings

- LIC / PO recovery & non recovery report
- DD / Cheque list
- Thrift Society Recovery details

YEARLY JOBS

The yearly transactions are invoked using this yearly jobs menu. The activities under this item are classified as

- Data entry
- Processing Jobs
- Reports

Data Entry

Data Entry at the beginning of the year involves entry of calender and week details. The calender details include all the months for the year along with the maximum days in each month and the festival and national holidays in each month. The details are entered in a table called CALENDER. The week details include the number of Sundays, Mondays etc., in every month in order to calculate the weekoff days of the employees. The days of the week are numbered 1 to 7 from Sunday through Saturday. These details are entered in the WEEKDET table.

Processing

This option is to initialize the EMPMST table for all the employees. The Credit CL days of the employees are updated for the year according to the employee's grade and category.

Reports

The reports taken at the end of the year are for ESI and PF respectively called Form - 6 and Form - 6A respectively.

MASTERS MAINTENANCE

This menu is for maintaining all the master tables. Adding of records, Modification of records, Deleting records and Inquiry of records are done as maintenance jobs. The various master tables are as follows

Control Master table

This master table is very important in terms of pay calculation. This table consists of only one record without any primary key. Entries into this table are to be made with utmost care.

This table contains details such as the fixed index point for DA calculation, the current month index point, the amount per index point, the Fixed DA amount, ESI Percentage, EPF Percentage, FPF Percentage, ESI rounding off factor, FPF maximum amount, PF rounding off factor, Thrift Society subscription amount, Emergency fund subscription amount, bus amount per trip, No. of bus trips for the month, the Operator Take Home Salary Percentage, the Staff Take Home Salary Percentage, Net pay rounding off factor, Pay month, Pay year and the year end date.

Department Master Table

The DEPT table is called the Department Master Table. It consists of the department code, department name and division. The department code is the primary key in this table.

Category Master Table

The CATEGORY table contains the category code, the category description, an indicator for Staff/Operator, an indicator for Confirmed/Not Confirmed and another indicator for Field Staff/Others. The category code is the primary key in this table.

Designation Master Table

The GRADE table is called the designation or grade master. It contains the grade code, Technical/Non Technical Indicator and the designation or grade description. The grade code and the technical indicator form the primary key for this table.

Cost Centre Master Table

The CSTCNTR table contains the cost centre code and the cost centre name. The cost centre code is the primary key in this table.

Pay Point Master Table

The PAYPNT table contains the pay point code and the pay point name. The pay point code is the primary key.

Union Master Table

The UNIONDET table contains the union indicator, the name of the union and the subscription amount for the particular union. The union id should be unique.

Pay Structure Master Table

The PAYSTRU table contains the pay details for a particular category and grade. The category and grade code form the primary key for this table. The details include basic minimum, maximum basic, increment amount, shift allowance, attendance bonus, medical allowance, education allowance, LTA, CCA, HRA, Vehicle allowance etc.

Employee Master Table

EMPMST table is an important table which contains all the related and relevant information about an employee like code, name, father's/husband's name, sex, date of birth, department code, date of joining, category, grade, basic pay, weekly off, Thrift Society membership indicator, ESI indicator, the Credit CL days and paise balance for the employee, the pay point code, the geographical indicator, etc., The employee code is the field which uniquely identifies a record.

Major Deduction Category Master

The DMAJCAT table contains the major deduction category code and the description. The major category code is the primary keyfield.

Deduction Sub Category Master

The DSUBCAT master table contains the major deduction category code, the deduction sub code, the description, the priority of the deduction, the Partial/Full indicator, the Amount/Percentage indicator, etc. The deduction major and sub codes form the primary key.

LIC-PO Master Table

The LICPODET table contains the employee code, the policy number, the LIC/PO indicator and the amount to be deducted monthly. For Post Office savings the accumulation amount is also maintained.

Professional Tax Master Table

The PTDET table contains the slab number, slab id, the minimum taxable amount, the maximum taxable amount and the tax amount per month. The slab number and slab id together form the primary key for this table.

Deduction from Bonus Master Table

The BONDET table contains the bonus year, employee code, deduction major category, sub category and the amount. The table can be accessed only by the combination of bonus year, employee code, deduction major category and sub category.

Cost Centre Percentage Master

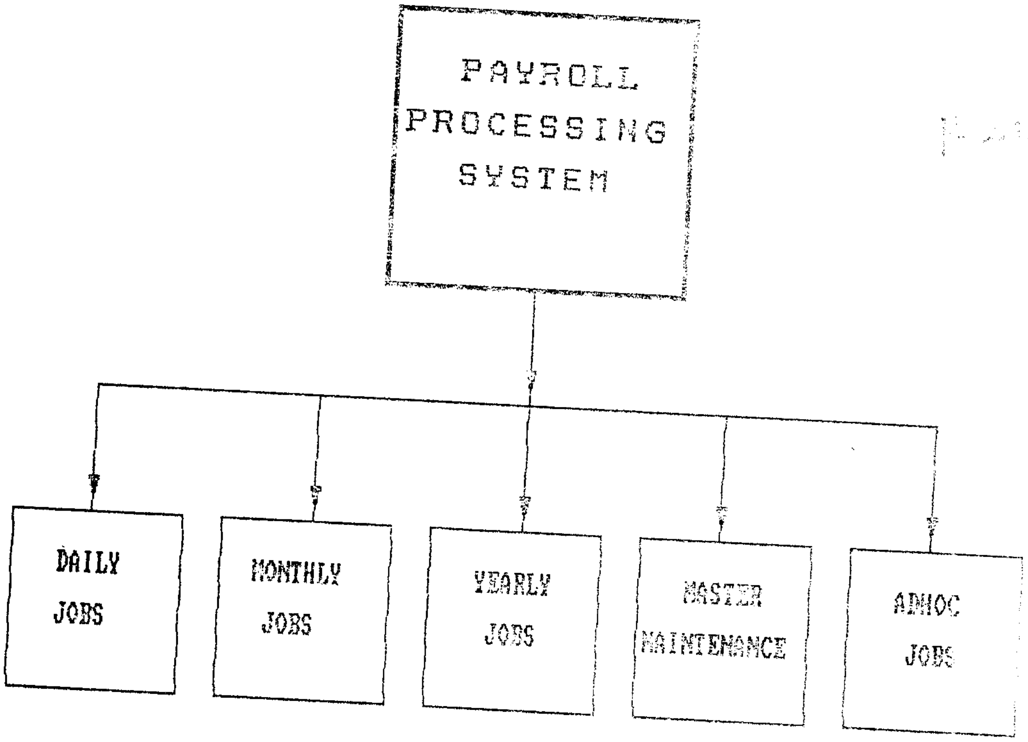
The CSTCNTRP table contains details like employee code, cost centre code and the cost percentage. The employee code and cost centre code uniquely identify a record from this table.

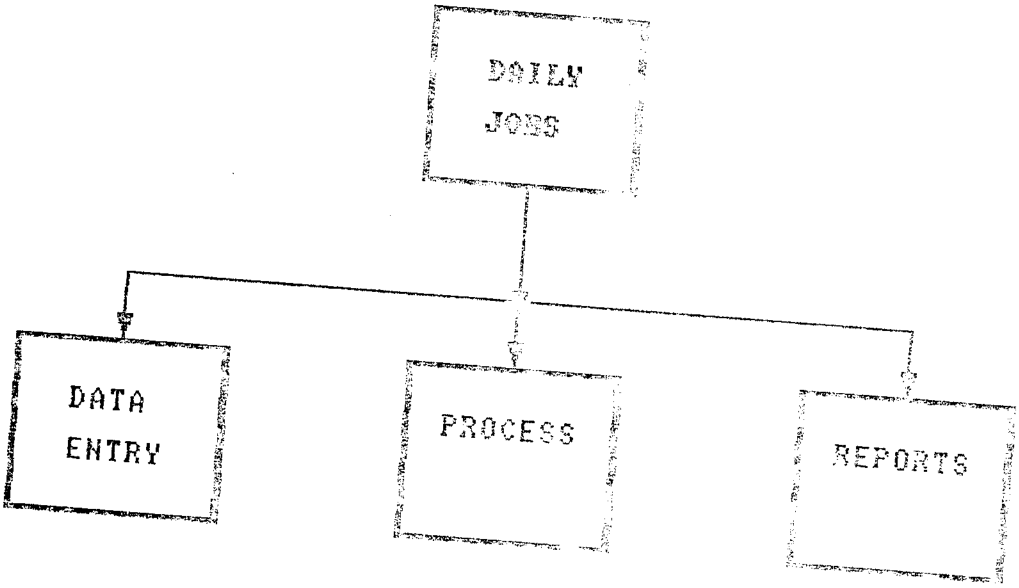
ADHOC JOBS

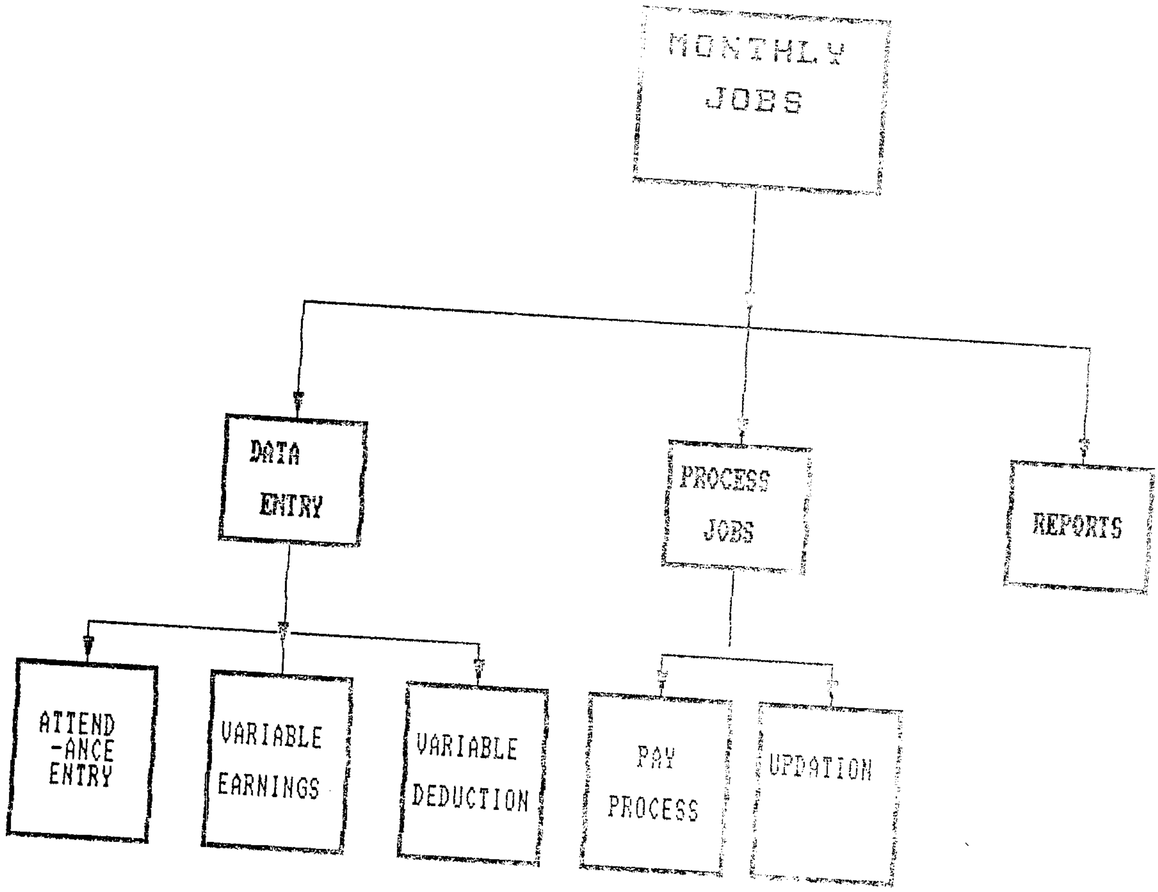
This involves a provision for taking backups. Backup can be taken for the whole database or any particular table at any point of time.

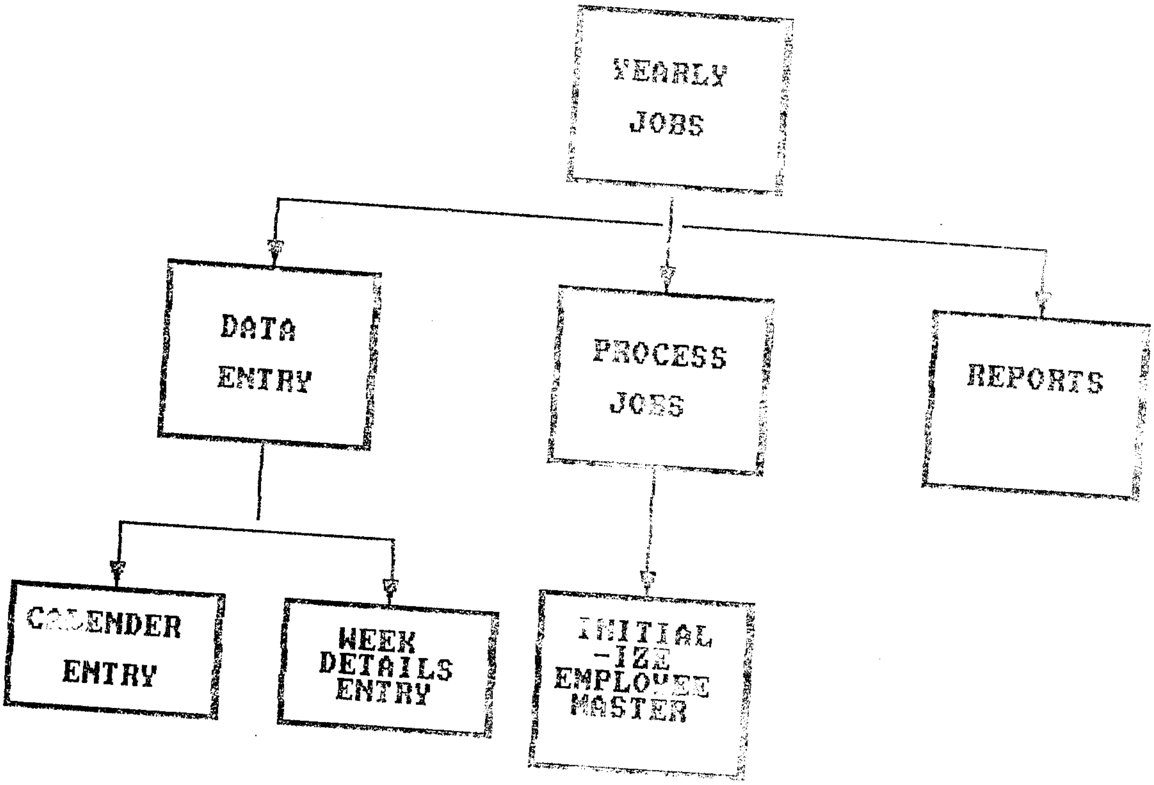
PAYROLL PROCESSING SYSTEM

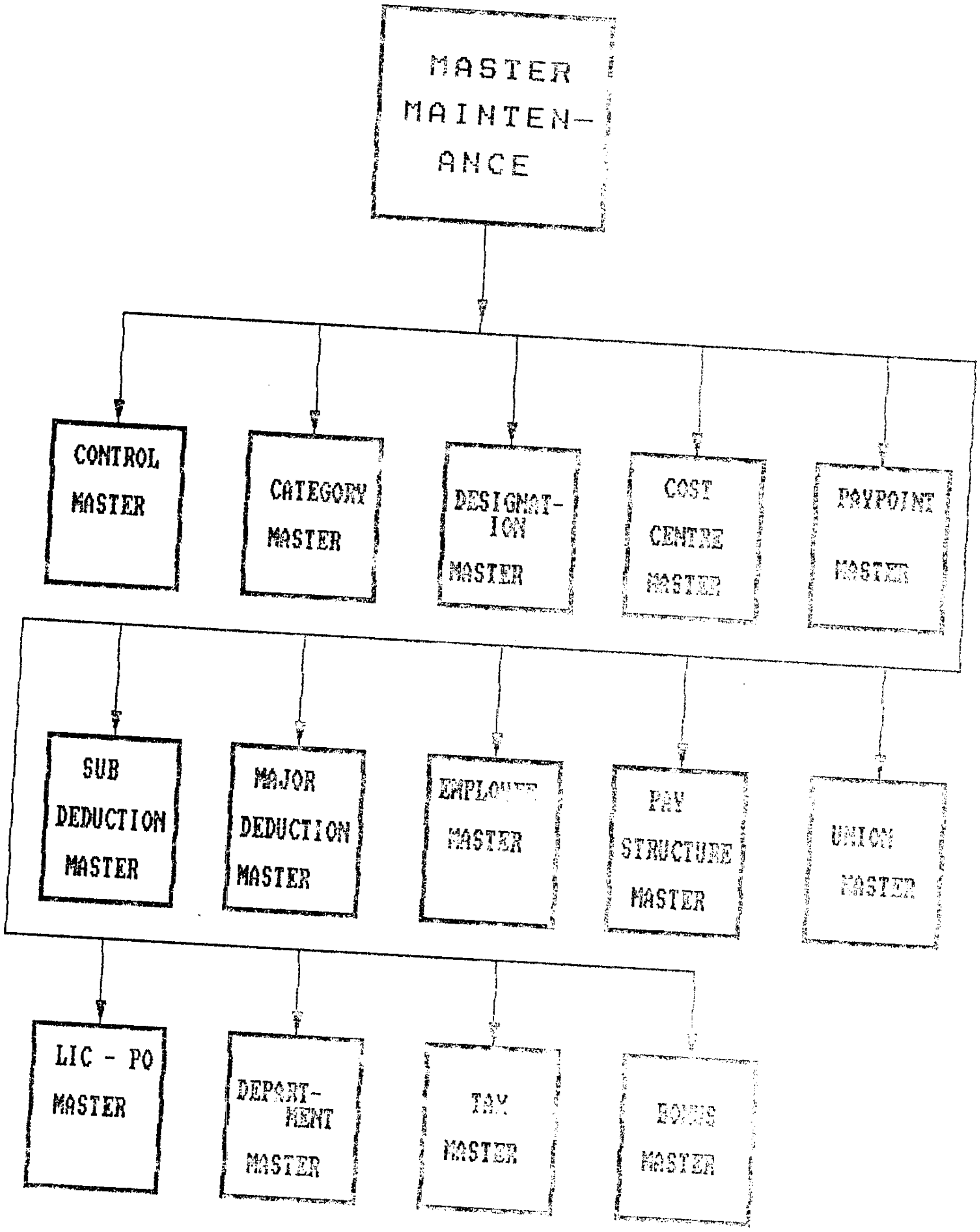
STRUCTURE CHART

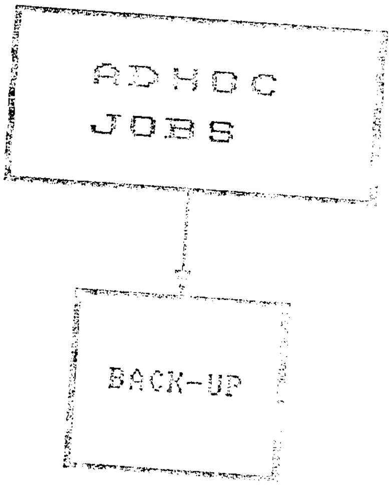



DAILY JOBS
DATA ENTRY
PROCESS
REPORTS

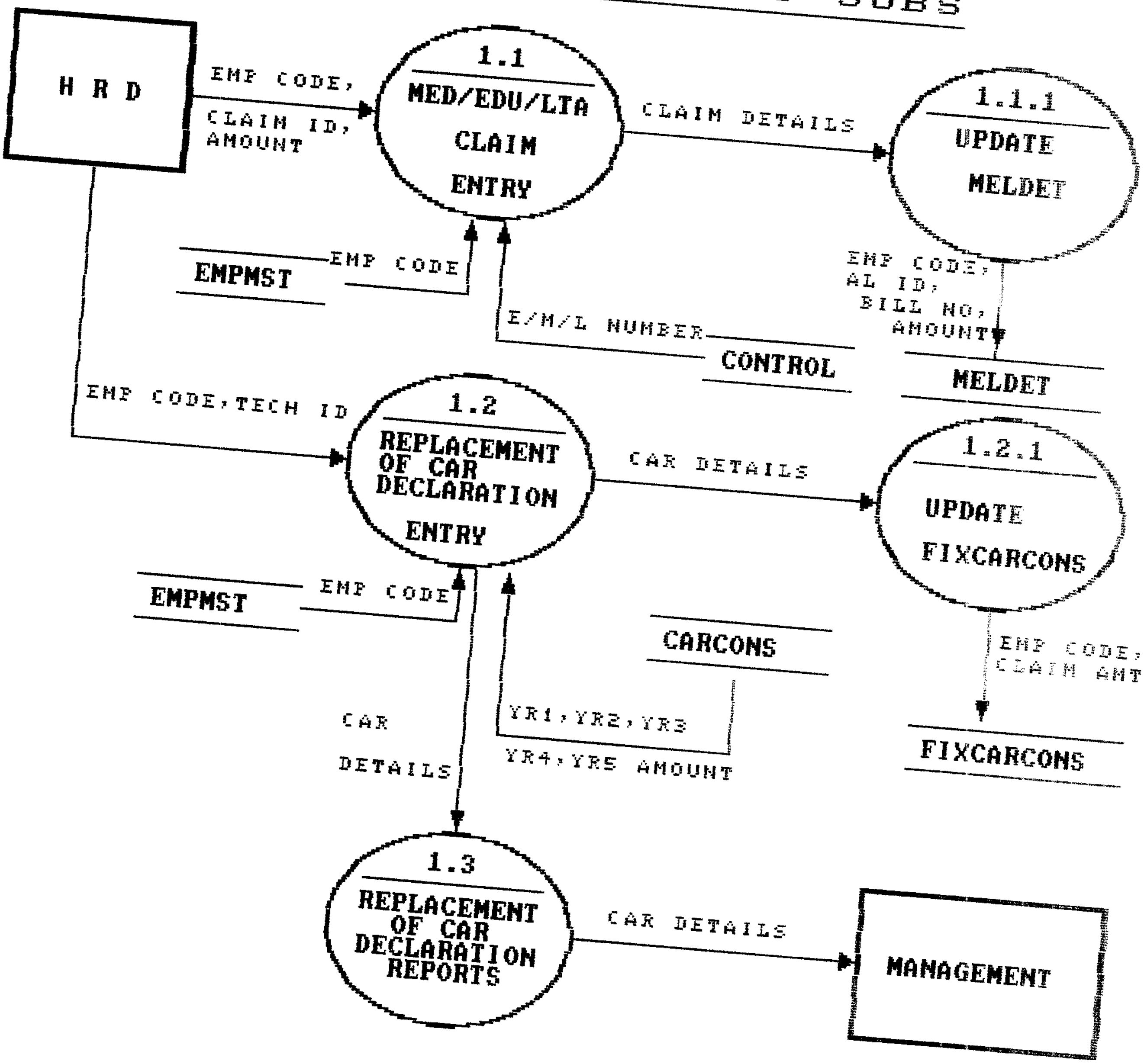

MONTHLY JOBS
DATA ENTRY
PROCESS JOBS
REPORTS
ATTEND -ANCE ENTRY
VARIABLE EARNINGS
VARIABLE DEDUCTION
PAY PROCESS
UPDATION


YEARLY JOBS
DATA ENTRY
PROCESS JOBS
REPORTS
CALENDER ENTRY
WEEK DETAILS ENTRY
INITIAL -IZE EMPLOYEE MASTER

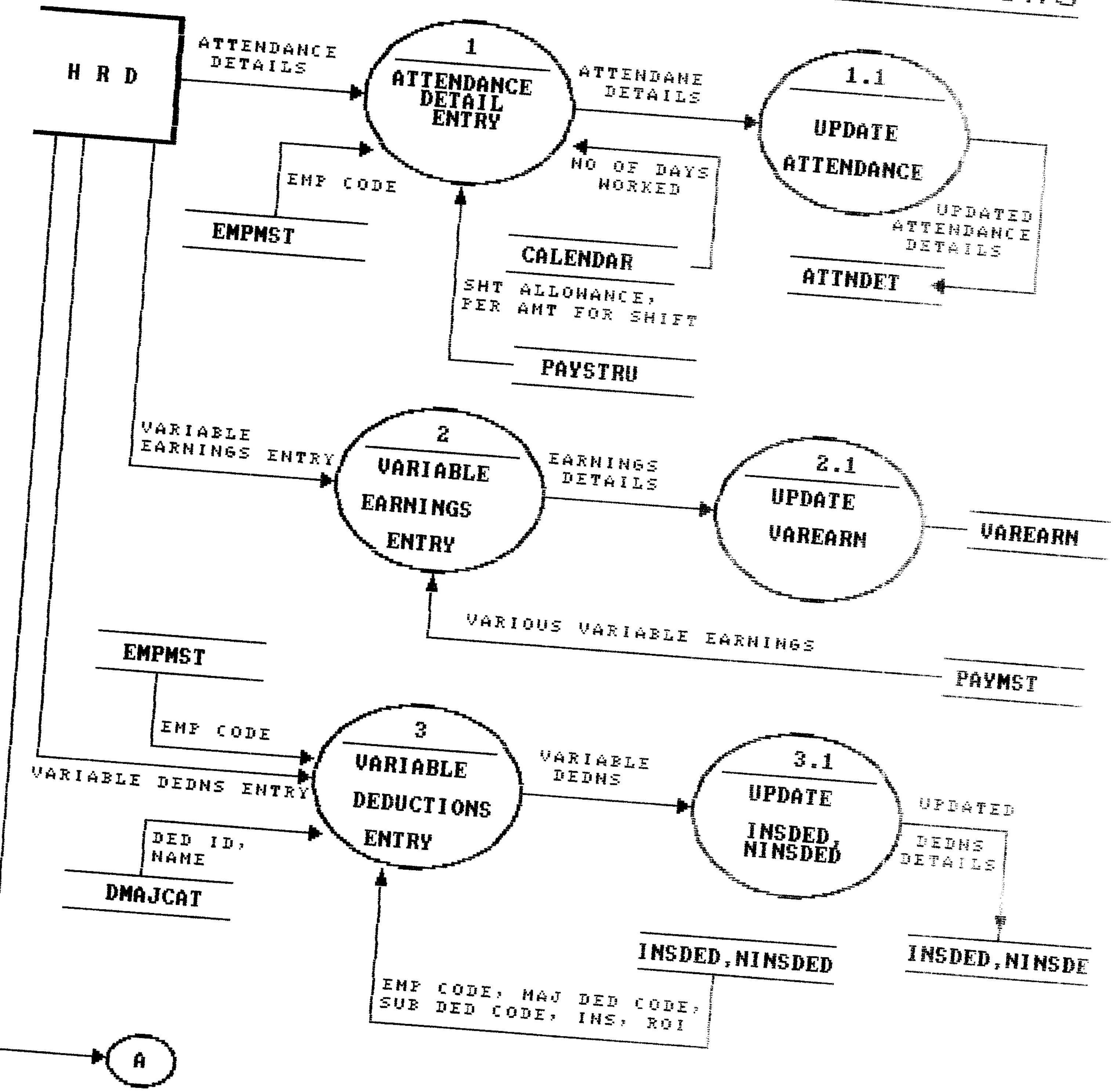

MASTER MAINTEN- ANCE
CONTROL MASTER
CATEGORY MASTER
DESIGNAT- ION MASTER
COST CENTRE MASTER
PAYPOINT MASTER
SUB DEDUCTION MASTER
MAJOR DEDUCTION MASTER
EMPLOYEE MASTER
PAY STRUCTURE MASTER
UNION MASTER
LIC - PO MASTER
DEPART- MENT MASTER
TAX MASTER
BONUS MASTER

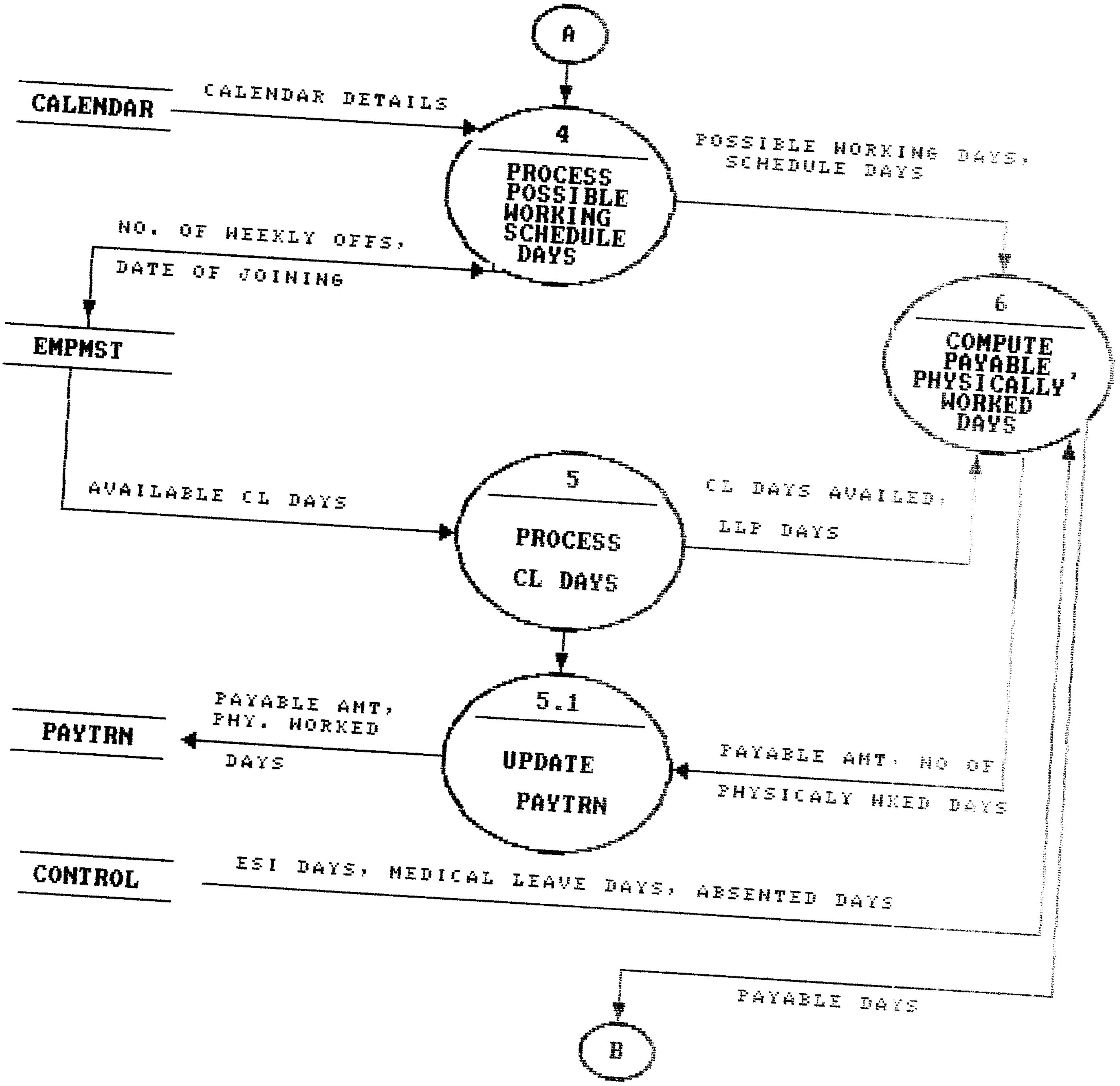

ADHOC
JOBS
BACK-UP

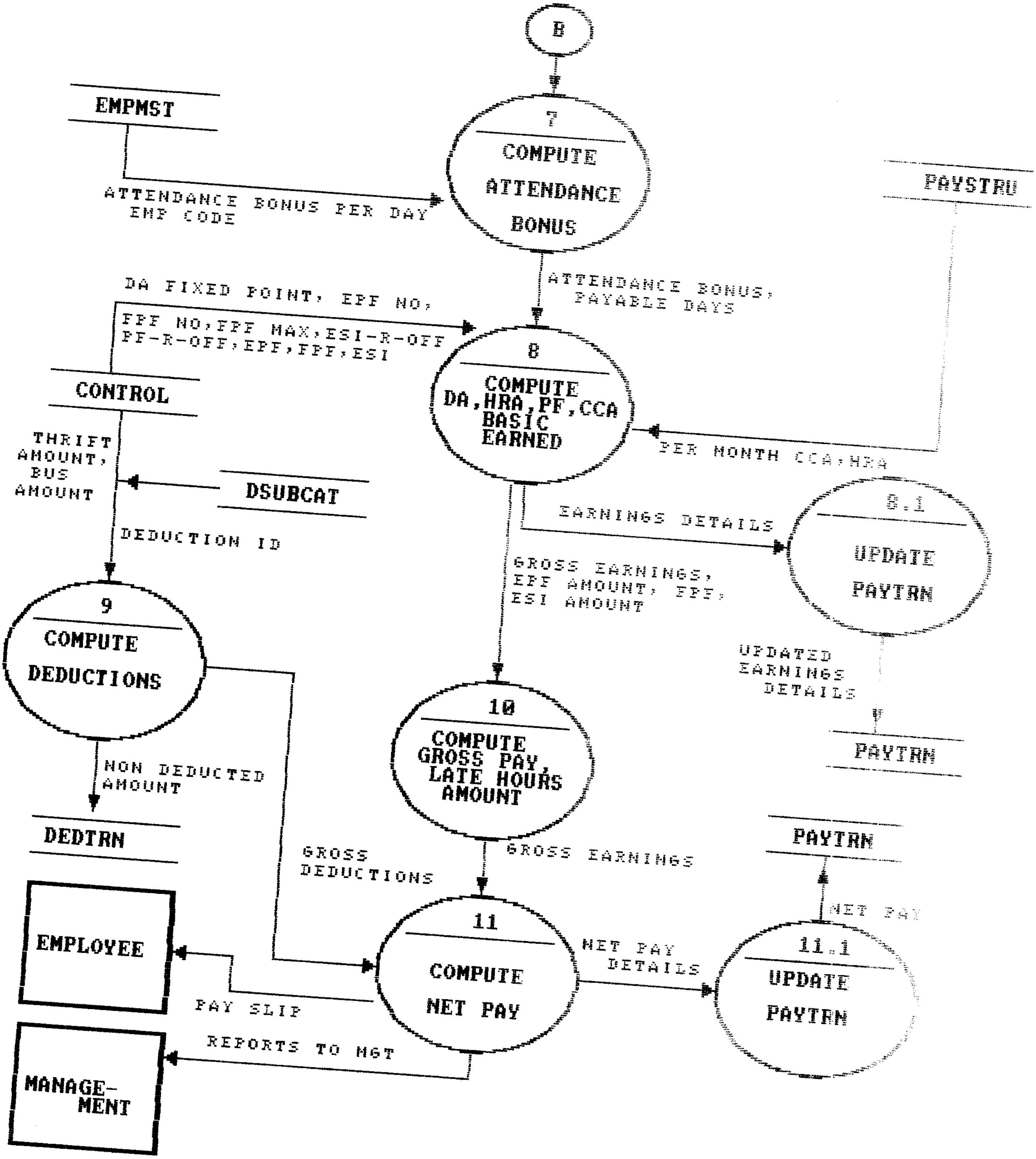
DFD FOR DAILY JOBS



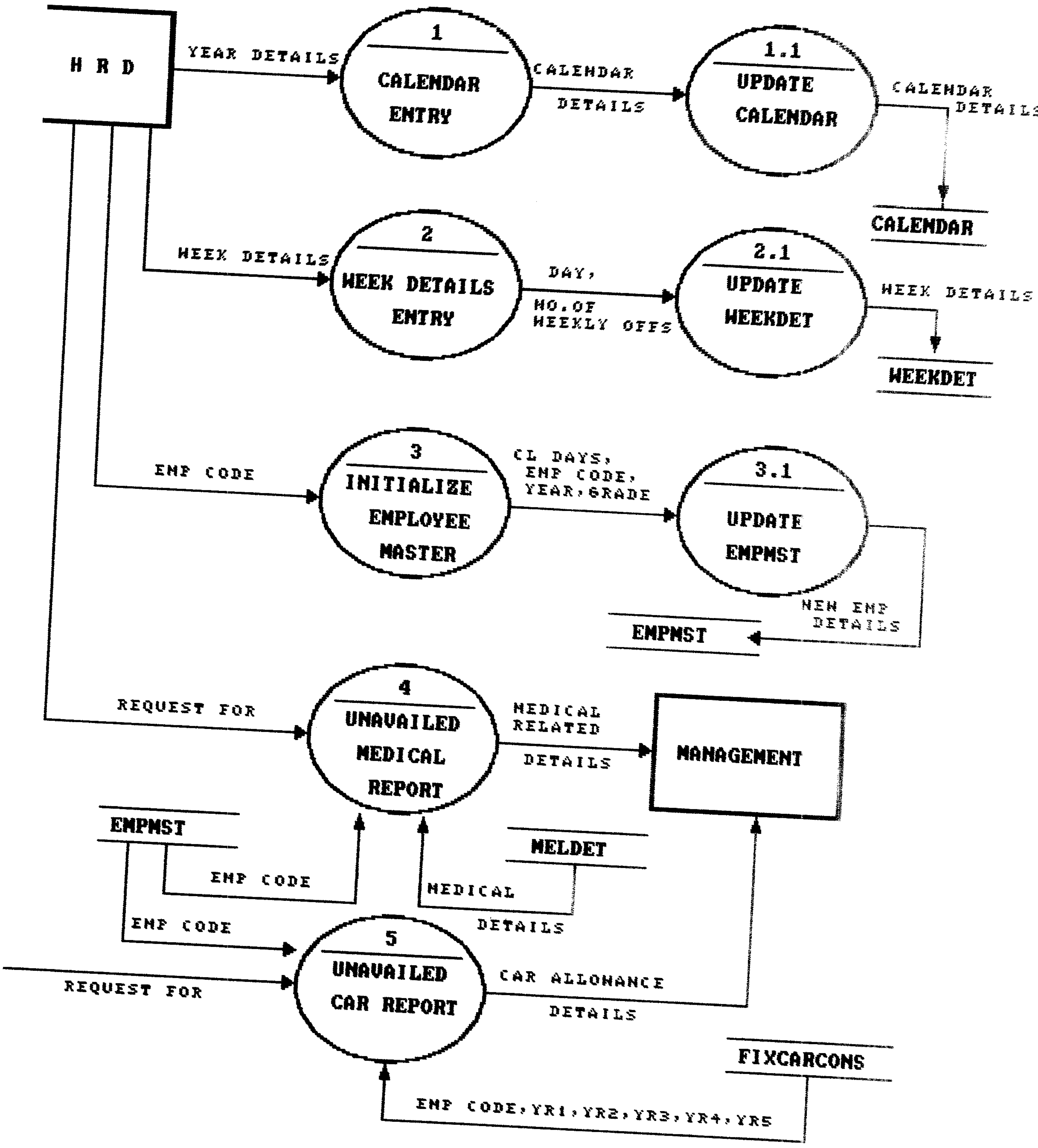
DFD FOR MONTHLY JOBS PROCESSING




A
CALENDAR
CALENDAR DETAILS
4
PROCESS POSSIBLE WORKING SCHEDULE DAYS
POSSIBLE WORKING DAYS, SCHEDULE DAYS
NO. OF WEEKLY OFFS,
DATE OF JOINING
EMPMST
6
COMPUTE PAYABLE PHYSICALLY' WORKED DAYS
AVAILABLE CL DAYS
5
PROCESS CL DAYS
CL DAYS AVAILED,
LLP DAYS
5.1
UPDATE PAYTRN
PAYTRN
PAYABLE AMT, PHY. WORKED DAYS
PAYABLE AMT, NO OF PHYSICALY WKED DAYS
CONTROL
ESI DAYS, MEDICAL LEAVE DAYS, ABSENTED DAYS
PAYABLE DAYS
B


B
EMPMST
7
COMPUTE ATTENDANCE BONUS
ATTENDANCE BONUS PER DAY
EMP CODE
PAYSTRU
ATTENDANCE BONUS,
PAYABLE DAYS
DA FIXED POINT, EPF NO,
FPF NO,FPF MAX,ESI-R-OFF
PF-R-OFF,EPF,FPF,ESI
8
COMPUTE DA,HRA,PF,CCA BASIC EARNED
CONTROL
PER MONTH CCA,HRA
THRIFT AMOUNT,
BUS AMOUNT
DSUBCAT
EARNINGS DETAILS
8.1
UPDATE PAYTRN
DEDUCTION ID
GROSS EARNINGS,
EPF AMOUNT, FPF,
ESI AMOUNT
9
COMPUTE DEDUCTIONS
UPDATED
EARNINGS
DETAILS
10
COMPUTE GROSS PAY, LATE HOURS AMOUNT
PAYTRN
NON DEDUCTED
AMOUNT
DEDTRN
GROSS
DEDUCTIONS
GROSS EARNINGS
PAYTRN
EMPLOYEE
11
COMPUTE NET PAY
NET PAY
NET PAY
DETAILS
11.1
UPDATE PAYTRN
PAY SLIP
REPORTS TO MGT
MANAGE-
MENT

DFD FOR YEARLY PROCESSING



1. CONTROL MASTER TABLE

```
print 'Creating Tables for Payroll Application'
go
print 'Creating Table control '
go
create table control
(
fidxpnt       smallint   not null, /*  Fixed Index Point                  */
cidxpnt       smallint   not null, /*  Current Month Index Point          */
pidxpntamt    money      not null, /*  Per Index Point Amount             */
fdaamt        money      not null, /*  Fixed DA Amount                    */
esiper        money      not null, /*  ESI Percentage                     */
epfper        money      not null, /*  EPF Percentage                     */
fpfper        money      not null, /*  FPF Percentage                     */
maxfpfamt     money      not null, /*  FPF Maximum Amount                 */
esiroff       money      not null, /*  ESI Rounding Off Factor            */
epfroff       money      not null, /*  EPF Rounding Off Factor            */
esilmt        smallint   not null, /*  ESI Limit                          */
efdsramt      money      not null, /*  Emergency Fund Subscription        */
                                   /*  Amount                             */
tftsramt      money      not null, /*  Thrift Society Subscription        */
                                   /*  Amount                             */
ostpamt       money      not null, /*  Per Trip Amount for Bus            */
sfixtp        tinyint    not null, /*  Fixed Bus Trips Per Month          */
tper          money      not null, /*  Death Percentage                   */
simnt         tinyint    not null, /*  ESI Month                          */
othsper     money      not null, /*   Operator Take Home Salary          */
                                 /*   Percentage                         */
sthsper     money      not null, /*   Staff Take Home Salary             */
                                 /*   Percentage                         */
ntpyroff    tinyint    not null, /*   Net Pay Rounding Off Factor        */
yrend       datetime   not null, /*   Year End Date                      */
paymnt      tinyint    not null, /*   Pay Month                          */
payyr       smallint   not null, /*   Pay Year                           */
lwelfd      money      not null, /*   Labour Welfare Fund                */
medbno      int        not null, /*   Medical Bill Number                */
edubno      int        not null, /*   Education Bill Number              */
ltabno      int        not null, /*   LTA Bill Number                    */
psmesg      char(50)       null, /*   Pay Slip Message [Last Line]       */
)
go
```

2. DEPARTMENT MASTER TABLE

```
print  'Creating Table dept'
go
create table dept
(
 dcd            char(5)   not null, /* PK -Department Code          */
 dnm            char(20)  not null, /*    Department Name           */
 div            char(20)  not null  /*    Division                  */
)
go
print 'Creating index on Table dept'
go
create unique index dept_idx on dept(dcd)
```

3. CATEGORY MASTER TABLE

```
print 'Creating Table category'
go
create table category
(
 cacd           smallint  not null, /* PK -Category Code                  */
 cadesc         char(20)  not null, /*    Category Description            */
 cascro         char(1)   not null, /*    Staff/Operator Indicator        */
 cacorn         char(1)   not null, /*    Confirmed/Not Confirmed         */
                                    /*    Indicator                       */
 caforo         char(1)   not null, /*    Field Personnel / Others        */
)
print 'Creating index on Table category'
go
create unique index cat_idx on category(cacd)
```

4. GRADE MASTER TABLE

```
print 'Creating Table grade'
go
create table grade
(
 gcd              char(5)   not null, /* PK1-Grade Code                  */
 gid              char(1)   not null, /* PK2-Technical Indicator         */
                                      /*    T - Technical                */
                                      /*    N - Non Technical            */
 gdesc            char(20)  not null  /*    Grade Description            */
)
go
print 'Creating index on Table grade'
go
create unique index  grade_idx on grade(gcd, gid)
go
```

5. PAYPOINT MASTER TABLE

```
print 'Creating Table paypoint'
go
create table paypoint
(
 ppcd             char(2)   not null, /* PK -Pay Point Code              */
 ppdesc           char(20)  not null  /*    Pay Point Description        */
)
go
print 'Creating index on Table paypoint'
go
create unique index paypnt_idx on paypoint(ppcd)
```

6. UNIONDET MASTER TABLE

```
print 'Creating Table uniondet'
go

create table uniondet
(

unid            char(1)   not null, /* PK -Union Id                       */
undesc          char(20)  not null, /*      Union Description             */
unsubamt        money     not null  /*      Subscription Amount           */
)
go

print 'Creating index on Table uniondet'
go

create unique index union_idx on uniondet(unid)
go
```

7. EMPLOYEE MASTER TABLE

```
print 'Creating Table empmst'
go

create table empmst
(

ecd             int         not null, /* PK -Employee Code                  */
enm             char(40)    not null, /*      Employee Name                 */
efnm            char(40)        null, /*      Employee's Father/Husband Name */
esex            char(1)     not null, /*      Employee's Sex                */
dcd             char(5)     not null, /* FK -Department Code                */
cacd            smallint    not null, /* FK -Category Code                  */
gcd             char(5)     not null, /* FK -Grade Code                     */
ebas            money       not null, /*      Employee's Basic Pay          */
edoj         datetime   not null,  /*    Date of Joining                   */
edob         datetime   not null,  /*    Date of Birth                     */
edoc         datetime       null,  /*    Date of Confirmation              */
ewkoff       smallint   not null,  /*    Weekly Off Day                    */
epfno        int            null,  /*    PF Number                         */
eesino       int            null,  /*    ESI Number                        */
epsbal       money      not null,  /*    Paise Balance                     */
eesiid       char(1)    not null,  /*    Esi Indicator                     */
egeoid       char(1)    not null,  /*    Geographical Indicator            */
ppcd         char(2)    not null,  /* FK -Pay Point Code                   */
etechid      char(1)    not null,  /*    Technical Indicator               */
ebsmid       char(1)    not null,  /*    Bus Membership Indicator          */
etftmid      char(1)    not null,  /*    Thrift Society Membership ID      */
eefdmid      char(1)    not null,  /*    Emergency Fund Indicator          */
ehraoid      char(1)    not null,  /*    HRA Optional Indicator            */
eunid        char(1)    not null,  /*    Union Membership Indicator        */
elpid        char(1)    not null,  /*    LIC - Post Office Indicator       */
ehacpid      char(1)    not null,  /*    Handicapped  Indicator            */
ereid        char(1)    not null,  /*    Grade Retained Indicator          */
ecrcl        money      not null,  /*    Credit CL                         */
cccd         char(3)    not null,  /*    Cost Centre Code                  */
eperdayid    char(1)    not null,  /*    Percentage / Day Indicator        */
eatnbon      money      not null,  /*    Attendance Bonus                  */
estat        char(1)    not null,  /*    Status [A]ctive, [L]eft           */
edol         datetime       null,  /*    Date of Leaving                   */
elres        char(40)       null,  /*    Reason for Leaving                */
)
```

```
go
print 'Creating index on Table empmst'
go
create unique index emp_ind on empmst(ecd)
go
```

8. CALENDER MASTER TABLE

```
print 'Creating Table calender'
go
create table calender
(
yr          smallint  not null, /* PK1-Year                        */
mnt         tinyint   not null, /* PK2-Month                       */
maxday      tinyint   not null, /*    Maximum Days in the month    */
nthol       tinyint   not null, /*    National Holidays            */
fsthol      tinyint   not null, /*    Festival Holidays            */
fnday1      tinyint   not null, /*    Holiday 1                    */
fnday2      tinyint   not null, /*    Holiday 2                    */
fnday3      tinyint   not null, /*    Holiday 3                    */
)
go
print 'Creating index on Table calender'
go
create unique index cal_idx on calender(yr, mnt)
go
```

9. WEEK DETAILS MASTER TABLE

```
print 'Creating Table weekdet'
go
create table weekdet
(
yr              smallint  not null, /* PK1-Year                         */
mnt             smallint  not null, /* PK2-Month                        */
ewkoff          smallint  not null, /*    Weekly  Off                   */
nodays          smallint  not null  /*    No. of  days                  */
)
go
print 'Creating index on Table weekdet'
go
create unique index week_idx on weekdet(yr, mnt)
go
```

10. PAY STRUCTURE MASTER TABLE

```
print 'Creating Table paystru'
go
create table paystru
(
cacd            smallint  not null, /* PK1-Category Code                */
gcd             char(5)   not null, /* PK2-Grade Code                   */
minbas          money     not null, /*    Basic Minimum                 */
maxbas          money     not null, /*    Basic Maximum                 */
incamt          money     not null, /*    Increment Amount              */
hra             money     not null, /*    House Rent Allowance          */
veha            money     not null, /*    Vehicle Allowance             */
cca             money     not null, /*    CCA                                */
spanamt         money     not null, /*    Super Annuation Amount             */
meda            money     not null, /*    Medical Allowance                  */
edua            money     not null, /*    Educational Allowance              */
lta             money     not null, /*    LTA                                */
cara            money     not null, /*    Fixed Car Allowance                */
atnbon          money     not null, /*    Attendance Bonus                   */
sfta            money     not null, /*    Shift Allowance                    */
elgdays         smallint  not null, /*    Eligible Leave Days                */
bonamt          money     not null  /*    Bonus Amount                       */
)
go
print 'Creating index on Table paystru'
go
create index pay_idx on paystru(cacd, gcd)
go
```

11. DEDUCTION MAJOR CATEGORY MASTER TABLE

```
print 'Creating Table dmajcat'
go
create table dmajcat
(
dmcd            char(2)   not null, /* PK -Major Deduction Category Code  */
mcdesc          char(20)  not null  /*    Category Description            */

o
rint 'Creating index on Table dmajcat'
o
```

```
create unique index dmaj_idx on dmajcat(dmcd)
go
```

12. DEDUCTION SUB CATEGORY MASTER TABLE

```
print 'Creating Table dsubcat'
go

create table dsubcat
(
dmcd            char(2)    not null, /* PK1-FK Major Deduction Category     */
                                                   /*     Code
dscd            smallint   not null, /* PK2-Deduction Sub Category Code     */
dpty            smallint   not null, /*     Deduction Priority              */
scdesc          char(20)   not null, /*     Deduction Description           */
scid            char(1)    not null, /*     Full/Partial Indicator          */
lorn            char(1)    not null  /*     Loan/Non Loan/Always Indicator */
)
go
print 'Creating index on Table dsubcat'
go
create unique index dsub_idx on dsubcat(dmcd, dscd)
go
```

13. PROFESSIONAL TAX MASTER TABLE

```
print 'Creating Table ptdet'
go
create table ptdet
(
slbno           char(2)    not null, /* PK -Slab Number
                                                                            */
slabid          char(1)    not null, /*     Slab Indicator
                                                                            */
```

```
  minamt         money     not null, /*    Minimum Amount                       */
  maxamt         money     not null, /*    Maximum Amount                       */
  pmntamt        money     not null  /*    Professional Tax Amount Per          */
                                     /*    Month                                */
 )
 go
 print 'Creating index on Table ptdet'
 go
 create unique index ptdet_idx on ptdet(sibno)
 go
```

14. ATTENDANCE MASTER TABLE

```
print 'Creating Table attn'
go
create table attn
(
 yr             smallint  not null, /* PK1- Year                              */
 mnt            tinyint   not null, /* PK2- Month                             */
 ecd            int       not null, /* PK3-FK Employee Code                   */
 daywk          money     not null, /*      Days Worked                       */
 clday          money     not null, /*      Casual Leave Days                 */
 llpday         money     not null, /*      Leave Loss of Pay Days            */
 absday         money     not null, /*      Absent Days                       */
 esiday         money     not null, /*      ESI Leave Days                    */
 loffday        money     not null, /*      Lay Off Days                      */
 mlday          money     not null, /*      Medical Leave Days                */
 nsanday        money     not null, /*      Not Santioned Days                */
 comday         money     not null, /*      Compensation Days                 */
 latehrs        money     not null, /*      Late Hours                        */
)
go
print 'Creating index on Table attn'
go
create unique index attn_idx on attn(yr, mnt, ecd)
go
```

15. LIC - PO MASTER TABLE

```
print 'Creating Table licpodet'
go
create table licpodet
(
 ecd        int       not null, /* PK1-FK Employee Code                       */
 dmcd       char(2)   not null, /* PK2-Major Deduction Category Code          */
 dscd       int       not null, /* PK3-Deduction Sub Category Code            */
 lpno       int       not null, /* PK4-LIC Policy Number                      */
 lpid       char(1)   not null, /*    LIC/Post Office Indicator               */
 insamt     money     not null, /*    Amount                                  */
 bal        money     not null, /*    Balance Amount                          */
 afg        char(1)   not null  /*    Active/Non Active/Deleted ID            */
)
go
print 'Creating index on Table licpodet'
go
create unique index lic_idx on licpodet(ecd, dmcd, dscd)
go
```

16. INSTALLMENT BASED DEDUCTION MASTER TABLE

```
print 'Creating Table insded'
go
create table insded
(
ecd        int       not null, /* PK1-FK Employee Code                  */
dmcd       char(2)   not null, /* PK2-FK Major Dedn. Category Code      */
dscd       smallint  not null, /* PK3-FK Dedn. Sub Category Code        */
lno        int           null, /* PK4-Loan No.                          */
dt         datetime  not null, /*      Date of commencement             */
priamt     money     not null, /*      Principal Amount                 */
totins     smallint  not null, /*      Total No. of Instalments         */
roi        money         null, /*      Rate Of Interest                 */
fstinsamt  money     not null, /*      First Instalment Amount          */
reginsamt  money     not null, /*      Regular Instalment Amount        */
insded     smallint  not null, /*      No.of deducted Instalments       */
dedamt     money     not null, /*      Deducted Amount                  */
cinsamt    money     not null, /*      Instalment amount Deducted       */
                               /*      for current month                */
cinsno     smallint  not null, /*      Current month Instalment No.     */
afg        char(1)   not null  /*      Active/Non Active/Deleted        */
                               /*      Indicator                        */
)
go
```

16. NON INSTALLMENT BASED DEDUCTION MASTER TABLE

```
print 'Creating index on Table insded'
go
create unique index ins_idx on insded(ecd, dmcd, dscd, lno)
go
print 'Creating Table ninsded'
go
create table ninsded
(
  ecd         int        not null, /* PK1-FK Employee Code                  */
  dmcd        char(2)    not null, /* PK2-FK Major Dedn. Category Code      */
  dscd        smallint   not null, /* PK3-FK Dedn. Sub Category Code        */
  lno         int            null, /* PK4-FK Loan No.                       */
  dt          datetime   not null, /*        Date of Commencement           */
  dedamt      money      not null, /*        Deduction Amount               */
  afg         char(1)    not null  /*     Active/Non Active/Deleted         */
                                   /*        Indicator                      */
)
go
print 'Creating index on Table ninsded'
go
create unique index nins_idx on ninsded(ecd, dmcd, dscd, lno)
go
```

17. VARIABLE EARNINGS MASTER TABLE

```
print 'Creating Table varearn'
go
create table varearn
(
 yr             smallint  not null, /* PK1-Year                          */
 mnt            tinyint   not null, /* PK2-Month                         */
 ecd            int       not null, /* PK3-FK Employee Code              */
 nosft          money     not null  /*    Number of extra shifts worked  */
)
go
print 'Creating index on Table varearn'
go
create unique index var_idx on varearn(yr, mnt, ecd)
go
```

18. PAY TRANSACTION TABLE

```
print 'Creating Table paytrn'
go
create table paytrn
(
 yr             smallint  not null, /* PK1-Year                          */
 mnt            tinyint   not null, /* PK2-Month                         */
 ecd            int       not null, /* PK3-FK Employee Code              */
 gcd            char(5)   not null, /*    Grade Code                     */
 basic          money     not null, /*    Basic Pay                      */
 hra            money     not null, /*    House Rent Allowance           */
 fda            money     not null, /*    Fixed DA                       */
```

```
    vda         money     not null, /*   Variable DA                            */
    veha        money     not null, /*   Vehicle Allowance                      */
    cca         money     not null, /*   CCA                                    */
    atnbon      money     not null, /*   Attendance Bonus                       */
    sfta        money     not null, /*   Shift Allowance                        */
    nosft       money     not null, /*   Number of shifts worked                */
    addp        money     not null, /*   Paise accumulation added in            */
                                    /*   netpay                                 */
    meda        money     not null, /*   Medical Allowance                      */
    edua        money     not null, /*   Educational Allowance                  */
    earned1     money     not null, /*   PF Salary                              */
    loffamt     money     not null, /*   Layoff Amount                          */
    groearn     money     not null, /*   Gross Earnings                         */
    groded      money     not null, /*   Gross Deductions                       */
    psbal       money     not null, /*   Paise Balance                          */
    epfamt      money     not null, /*   EPF Amount                             */
    fpfamt      money     not null, /*   FPF Amount                             */
    esiamt      money     not null, /*   ESI Amount                             */
    netpay      money     not null, /*   Net Pay                                */
    updfg       char(1)   not null, /*   Update Flag                            */
    payday      money     not null  /*   Payable Days                           */
)
go
print 'Creating index on Table paytrn'
go
create unique index paytrn_idx on paytrn(yr, mnt, ecd)
go
```

19. DEDUCTION TRANSACTION TABLE

```
print 'Creating table dedtrn'
go
create table dedtrn
(
yr          smallint  not null, /* PK1-Year                               */
mnt         tinyint   not null, /* PK2-Month                              */
ecd         int       not null, /* PK3-FK Employee Code                   */
dmcd        char(2)   not null, /* PK4-FK Major Dedn.Category Code        */
dscd        smallint  not null, /* PK5-FK Dedn. Sub Category Code         */
lno         int           null, /* PK6-FK Loan No.                        */
dedamt      money     not null, /*        Deducted Amount                 */
ndedamt     money     not null  /*        Non Deducted Amount             */
)
go
print 'Creating index on Table dedtrn'
go
create unique index dedtrn_idx on dedtrn(yr, mnt, ecd, dmcd, dscd, lno)
go
```

20. COST CENTRE MASTER TABLE

```
print 'Creating table cstcntr'
go
create table cstcntr
(
cccd        char(3)   not null, /* PK -Cost centre code                   */
ccdesc      char(30)  not null  /*     Description                        */
)
go
```

```
print 'Creating index on table cstcntr'
go
create unique index cst_idx on cstcntr(cccd)
go
```

21. COST CENTRE PERCENTAGE MASTER TABLE

```
print 'Creating table cstper'
go
create table cstper
(
 ecd          int       not null, /* PK1-FK Employee code            */
 cccd         char(3)   not null, /* PK2-FK Cost centre code         */
 cper         money     not null  /*        Cost centre percentage   */
)
go
print 'Creating index on Table cstper'
go
create unique index cstp_idx on cstper(ecd, cccd)
go
```

22. BONUS DEDUCTION MASTER TABLE

```
print 'Creating table Bonus Deduction'
go
create table bondet
(
 bdyr         smallint  not null, /* PK1-Bonus deduction year        */
 ecd          int       not null, /* PK2-FK Employee code            */
 dmcd         char(2)   not null, /* PK3-FK Major deduction code     */
 dscd         smallint  not null, /* PK4-FK Deduction sub code       */
 bdamt              money      not null  /*     Bonus deduction amount          */
)
go
print 'Creating index on Table bondet'
go
create unique index bon_idx on bondet(bdyr, ecd, dmcd, dscd)
go
```

SOFTWARE DEVELOPMENT

The Payroll System was developed in SYBASE SYSTEM 4 under the UNIX environment. Various tools provided by SYBASE were used in the development of the Payroll System. The Database, Rules, Defaults and Triggers were created in ISQL. Stored procedures were written in Transact SQL. Data entry screens were designed using DB-Library and Application Productivity Tools (APT). APT-Tools namely APT-Edit, APT-Build and APT-Execute were also used. The Pay Processing Program was written using DB-Library with 'C' as the 3GL. 'C' has been chosen as the host language. C combines elements of high-level language with the functionalism of assembly language. Pointers are one of C's strongest features. Pointers support C's dynamic allocation routines. They improve the efficiency and speed of various routines. The ANSI C compiler was used. Payslips and reports are generated using DB-Library routines and a tool called Report Processor (RPT).

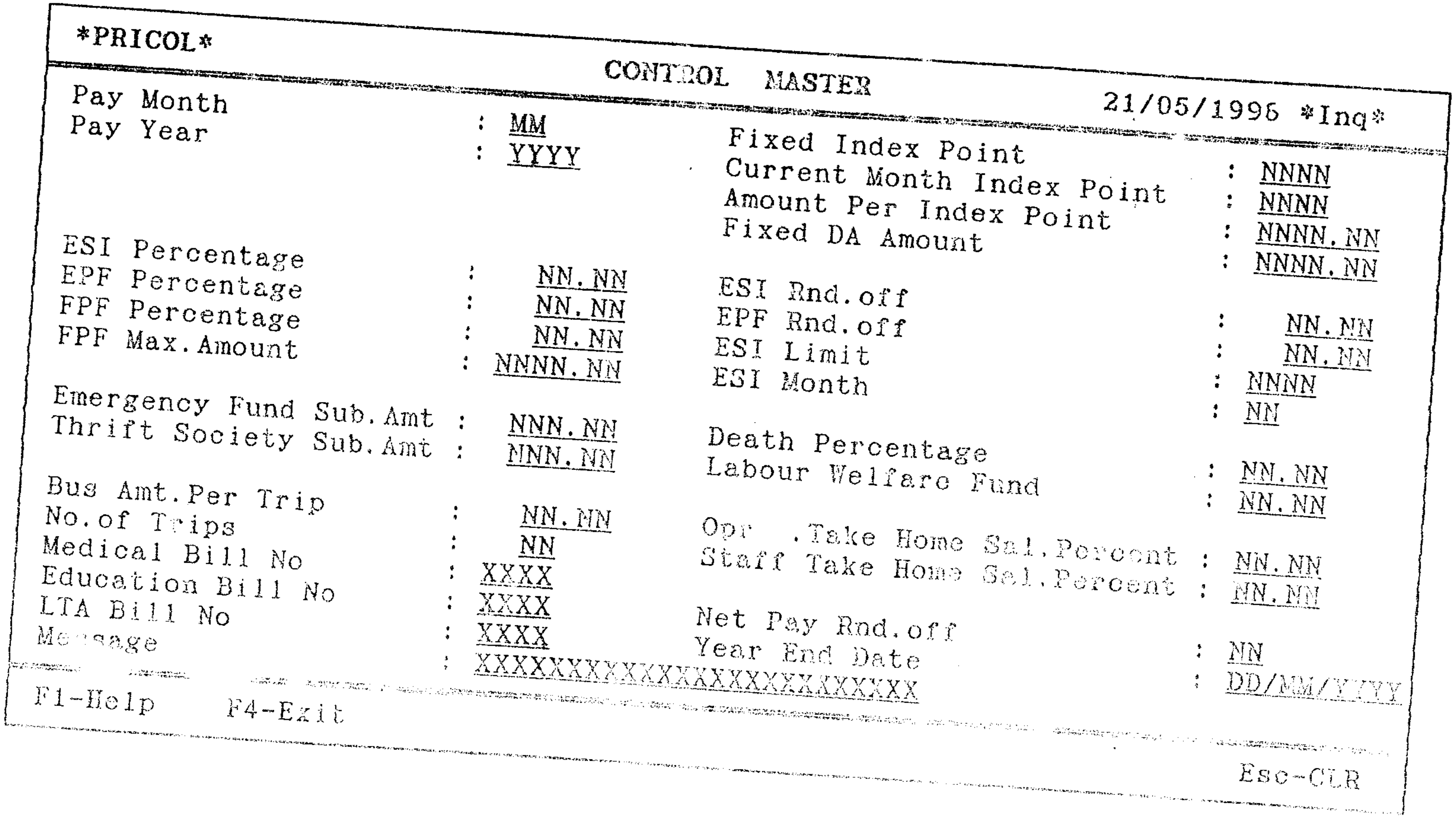
The System was developed in a HP 9000 Series 832 machine with HP-UX ver 9.04 as the operating system. It has a main memory of 64MB with an exclusive swap area of 61MB. The Hard Disk capacity of the machine is 2GB. The Risc processor used is 7100LC. It has a clock speed of 64MHZ. About 40 MB of disk space is reserved for the vast database. The machine has 40 terminals. The Payroll System will also be run at the Periyanayakkanpalayam factory on a HP 9000 series 817 machine with 70 terminals.

A study of the existing system was undertaken for a period of one month. Several drawbacks were identified in the existing system. The new payroll system was designed using software engineering tools. The system design was accepted by the organization. The coding was completed in about 3 months time. The new system was then run in parallel with the existing system. A few minor errors were rectified and the implementation work was completed.

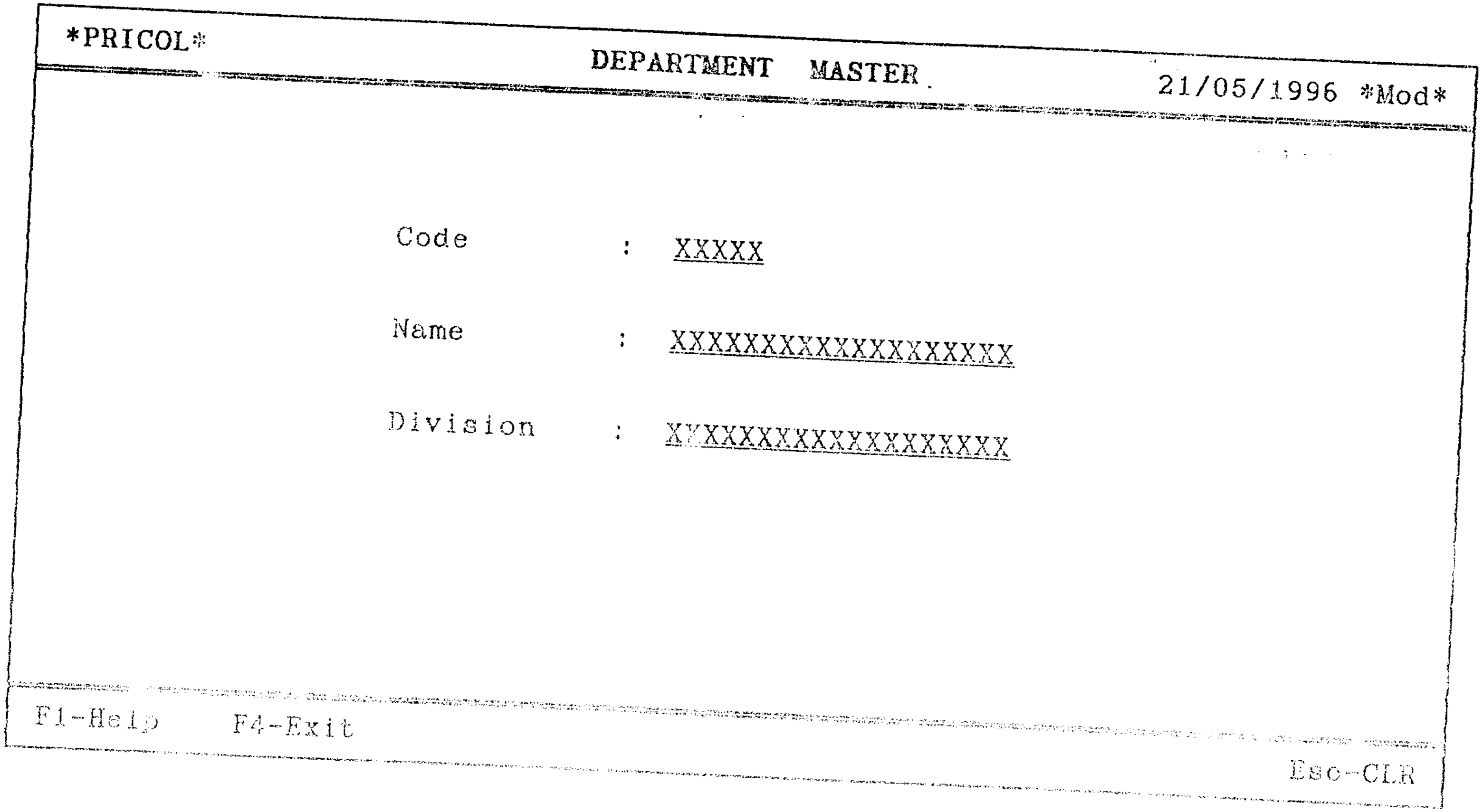
CONCLUSION AND PROJECTIONS

In the new Payroll System, the attendance and variable earnings details were entered in separate tables unlike in the old system where they were entered in the PAYTRN table. Hence this allows modification and checking of data in the respective tables and thus overcoming the necessity to re-run the pay process. SYBASE SQL Server processes 1000 rows per second. Thus it takes only 25 minutes to run the pay process for all the employees at PRICOL contrary to the old system which takes 3 days to complete the process. The advanced features of SYBASE has helped to overcome the drawbacks in the old system in GENESIS. All the design problems in the old system were solved in the new system. Thus a new system overcoming the existing drawbacks has been developed and implemented.

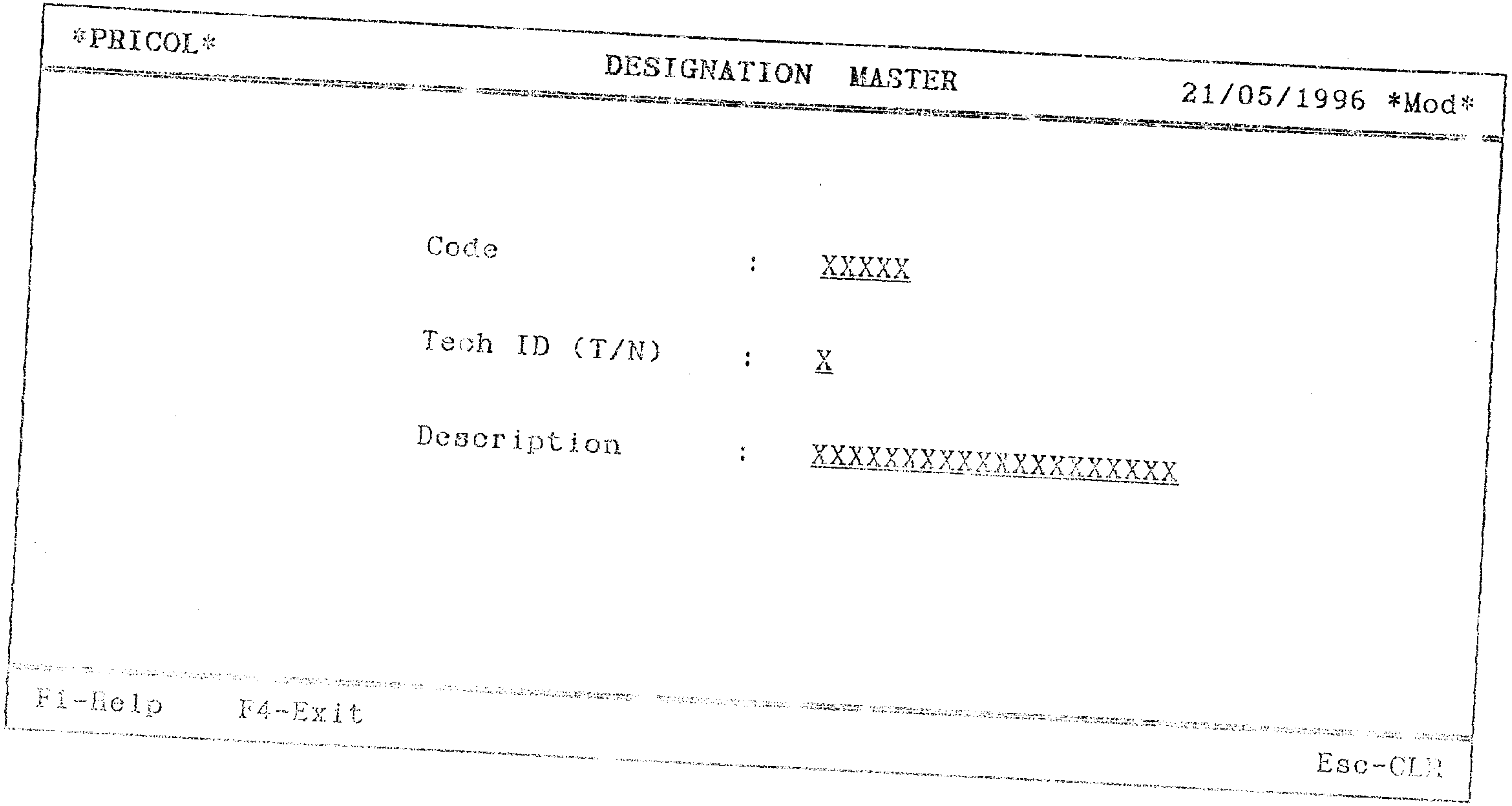
The New Payroll System can also be expanded in future under various modules. An interface can be made with the Financial Accounting System passing the net salary amount to be paid for the month, etc. The system can also be made generalized so that the design can be used for other similar organizations.



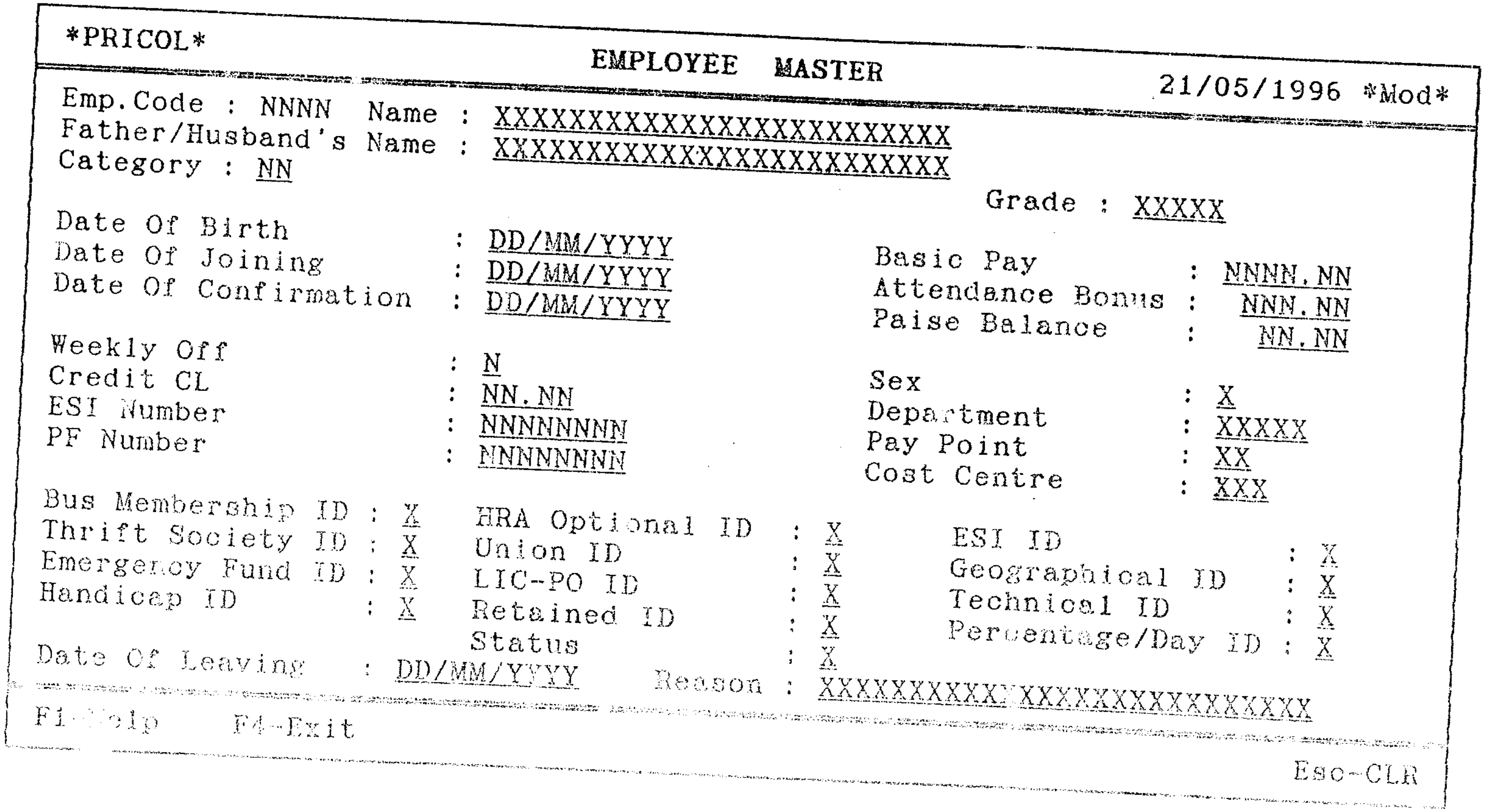
```
*PRICOL*
                              CONTROL  MASTER                          21/05/1996 *Inq*
Pay Month                   : MM         Fixed Index Point                  : NNNN
Pay Year                    : YYYY       Current Month Index Point          : NNNN
                                         Amount Per Index Point             : NNNN.NN
                                         Fixed DA Amount                    : NNNN.NN
ESI Percentage              :   NN.NN    ESI Rnd.off                        :   NN.NN
EPF Percentage              :   NN.NN    EPF Rnd.off                        :   NN.NN
FPF Percentage              :   NN.NN    ESI Limit                          : NNNN
FPF Max.Amount              : NNNN.NN    ESI Month                          : NN

Emergency Fund Sub.Amt      :  NNN.NN    Death Percentage                   : NN.NN
Thrift Society Sub.Amt      :  NNN.NN    Labour Welfare Fund                : NN.NN

Bus Amt.Per Trip            :   NN.NN    Opr  .Take Home Sal.Percent        : NN.NN
No.of Trips                 :   NN       Staff Take Home Sal.Percent        : NN.NN
Medical Bill No             : XXXX
Education Bill No           : XXXX       Net Pay Rnd.off                    : NN
LTA Bill No                 : XXXX       Year End Date                      : DD/MM/YYYY
Message                     : XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
F1-Help    F4-Exit
                                                                              Esc-CLR
```



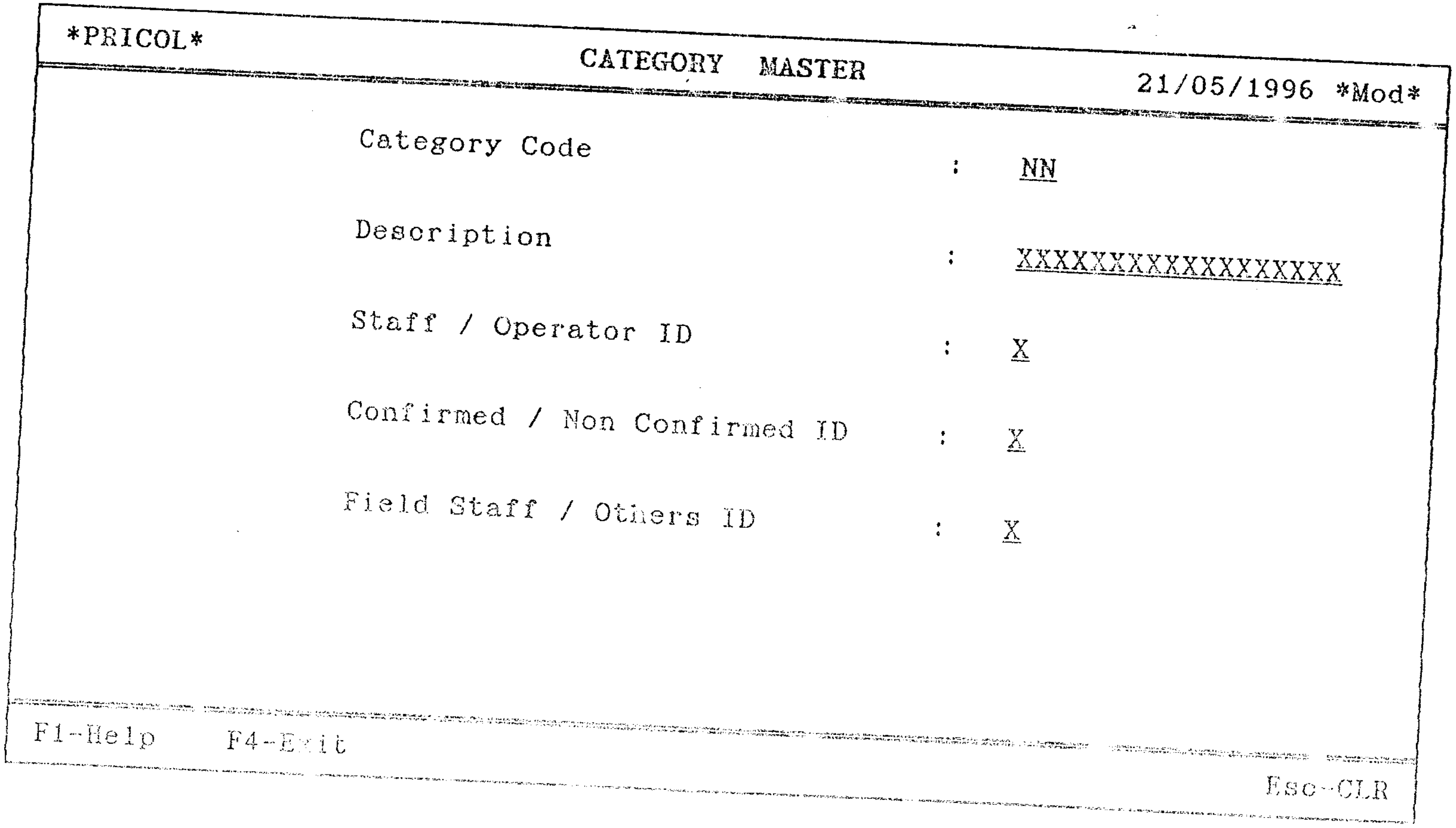
```
*PRICOL*                 DEPARTMENT  MASTER.            21/05/1996 *Mod*

              Code       :  XXXXX

              Name       :  XXXXXXXXXXXXXXXXXXX

              Division   :  XXXXXXXXXXXXXXXXXXX

F1-Help    F4-Exit
                                                                   Esc-CLR
```



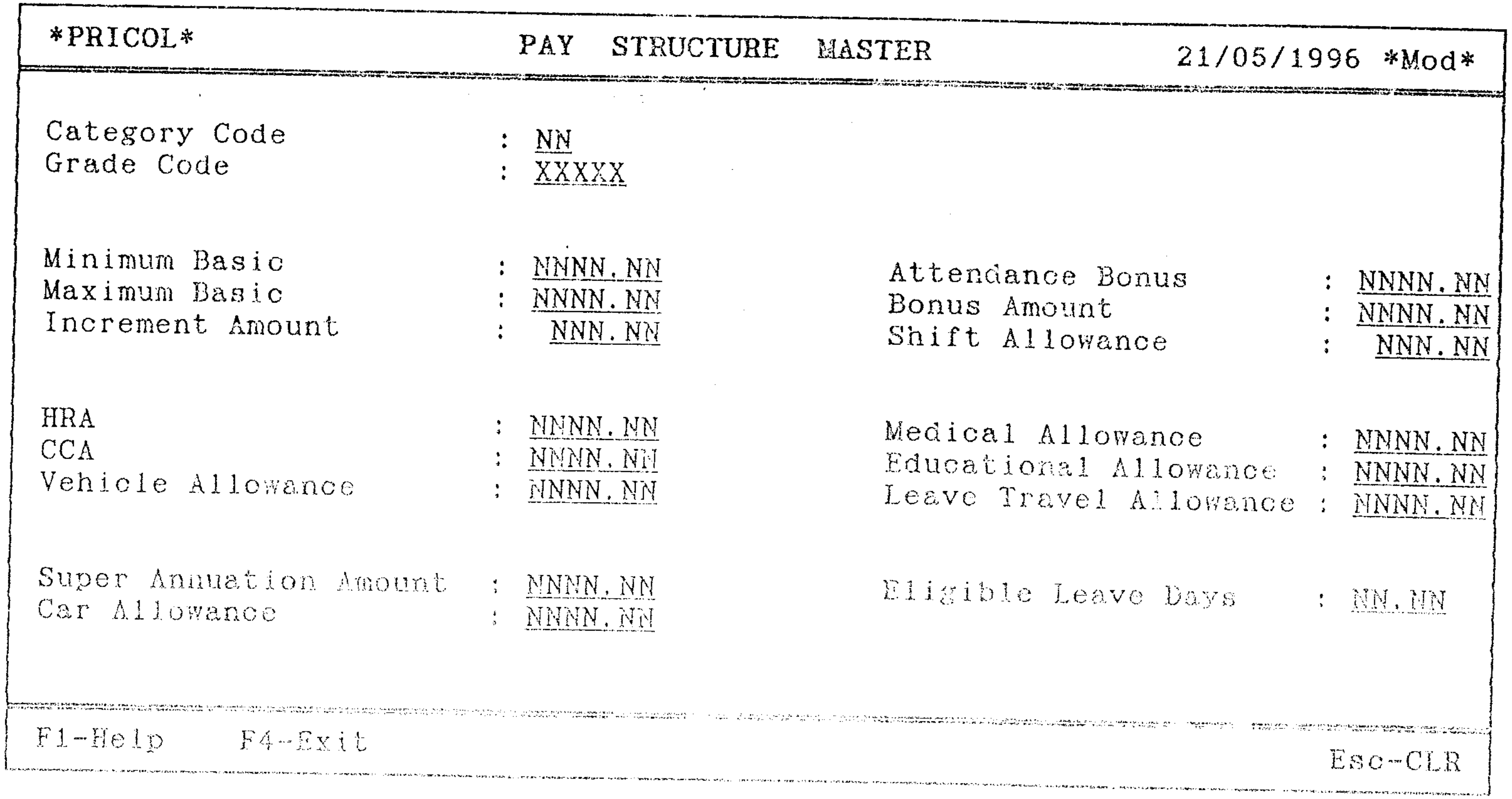
```
*PRICOL*
                         DESIGNATION  MASTER                 21/05/1996 *Mod*

                 Code                 :  XXXXX

                 Tech ID (T/N)        :  X

                 Description          :  XXXXXXXXXXXXXXXXXXXX

Fi-Help    F4-Exit
                                                                        Esc-CLR
```



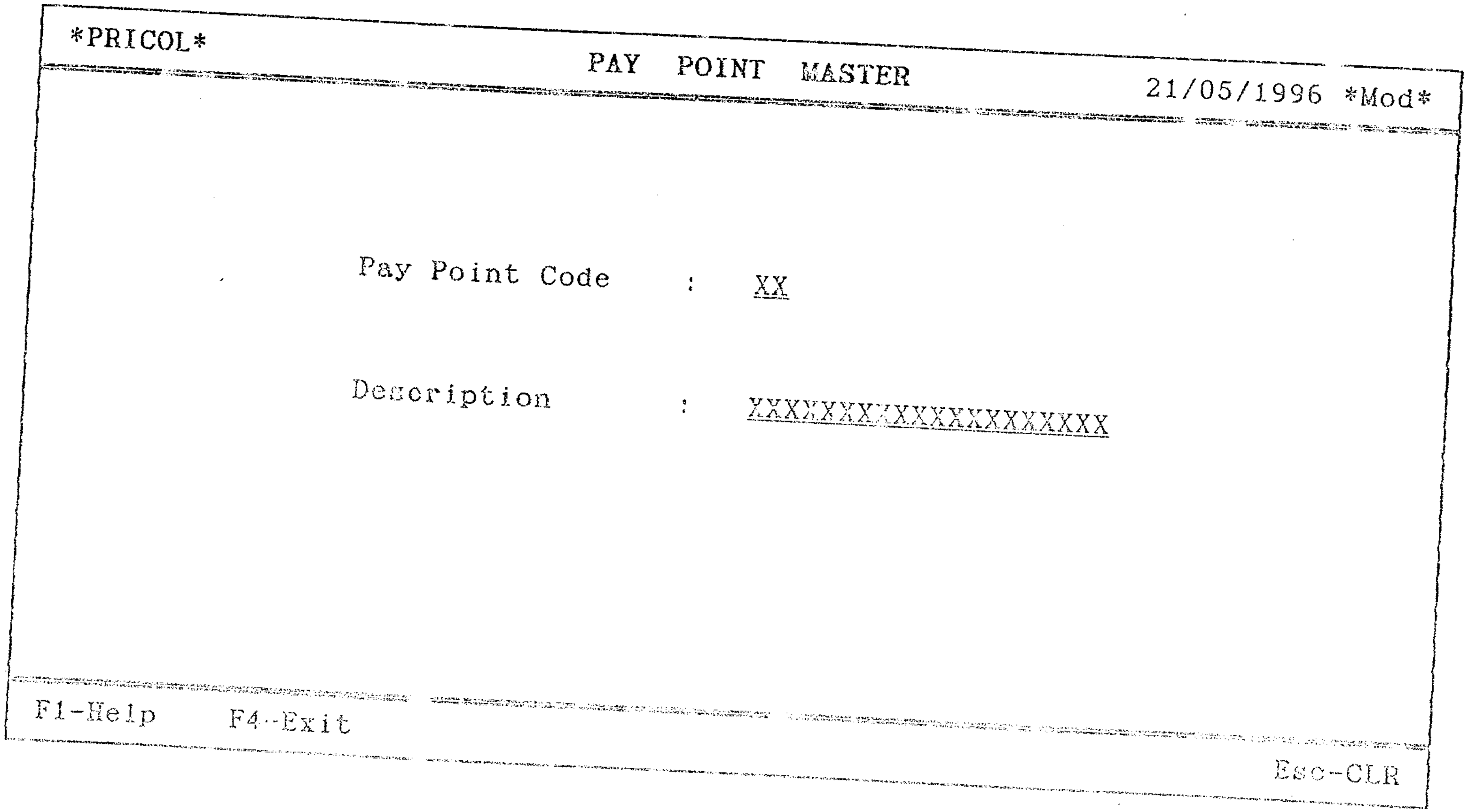
```
*PRICOL*
                          EMPLOYEE  MASTER
                                                              21/05/1996 *Mod*
Emp.Code : NNNN  Name : XXXXXXXXXXXXXXXXXXXXXXXXX
Father/Husband's Name : XXXXXXXXXXXXXXXXXXXXXXXXX
Category : NN
                                                 Grade : XXXXX
Date Of Birth            : DD/MM/YYYY
Date Of Joining          : DD/MM/YYYY         Basic Pay          : NNNN.NN
Date Of Confirmation     : DD/MM/YYYY         Attendance Bonus   :  NNN.NN
                                              Paise Balance      :   NN.NN
Weekly Off               : N
Credit CL                : NN.NN              Sex                : X
ESI Number               : NNNNNNNN           Department         : XXXXX
PF Number                : NNNNNNNN           Pay Point          : XX
                                              Cost Centre        : XXX
Bus Membership ID : X  HRA Optional ID : X
Thrift Society ID : X  Union ID        : X     ESI ID                 : X
Emergency Fund ID : X  LIC-PO ID       : X     Geographical ID        : X
Handicap ID       : X  Retained ID     : X     Technical ID           : X
                       Status          : X     Percentage/Day ID      : X
Date Of Leaving   : DD/MM/YYYY   Reason : XXXXXXXXXXXXXXXXXXXXXXXXX
F1-Help     F4-Exit
                                                                   Esc-CLR
```



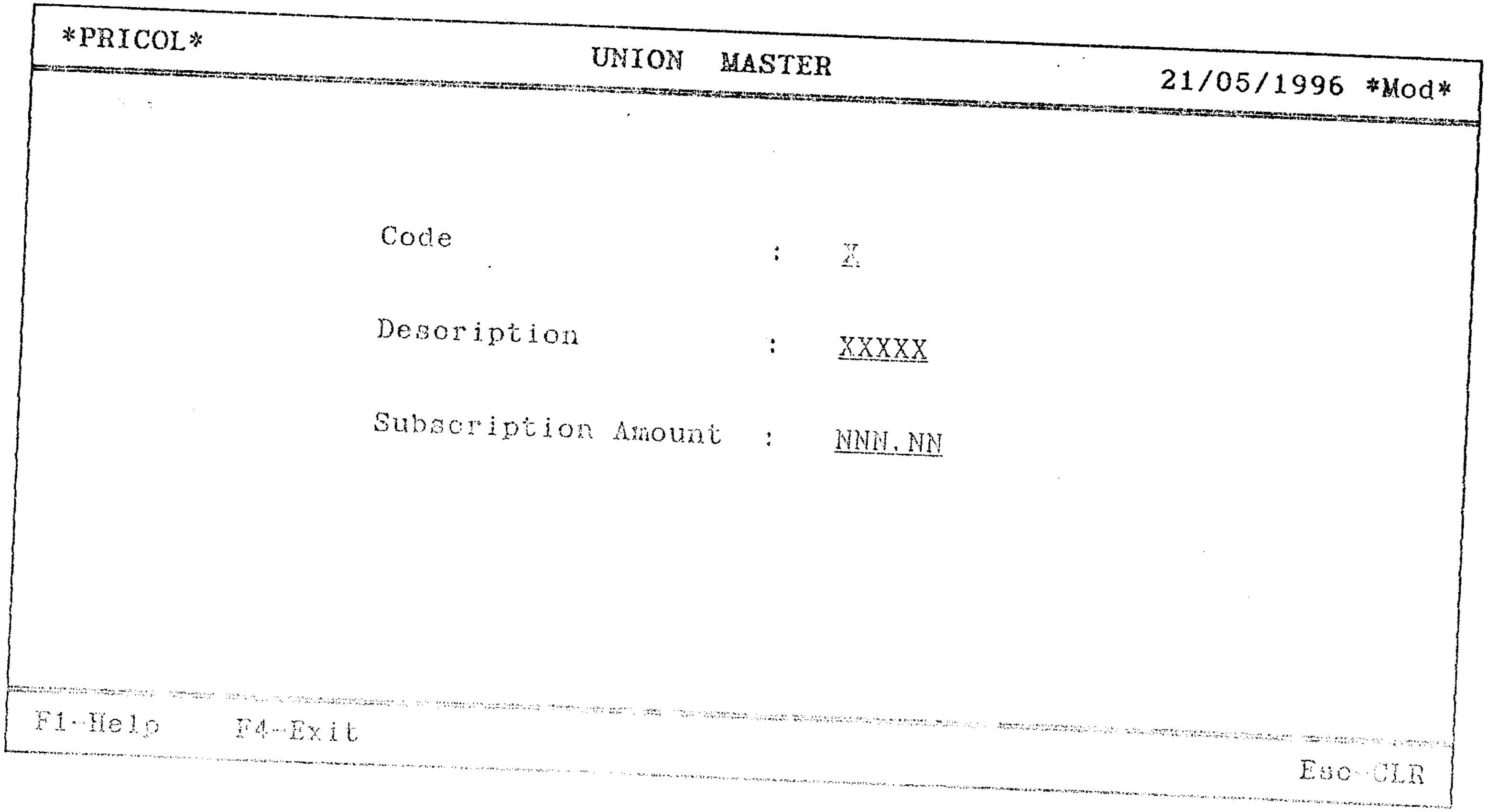
```
*PRICOL*
                        CATEGORY  MASTER                    21/05/1996 *Mod*

                Category Code                   :  NN

                Description                     :  XXXXXXXXXXXXXXXXXX

                Staff / Operator ID             :  X

                Confirmed / Non Confirmed ID    :  X

                Field Staff / Others ID         :  X

F1-Help    F4-Exit
                                                                  Esc-CLR
```

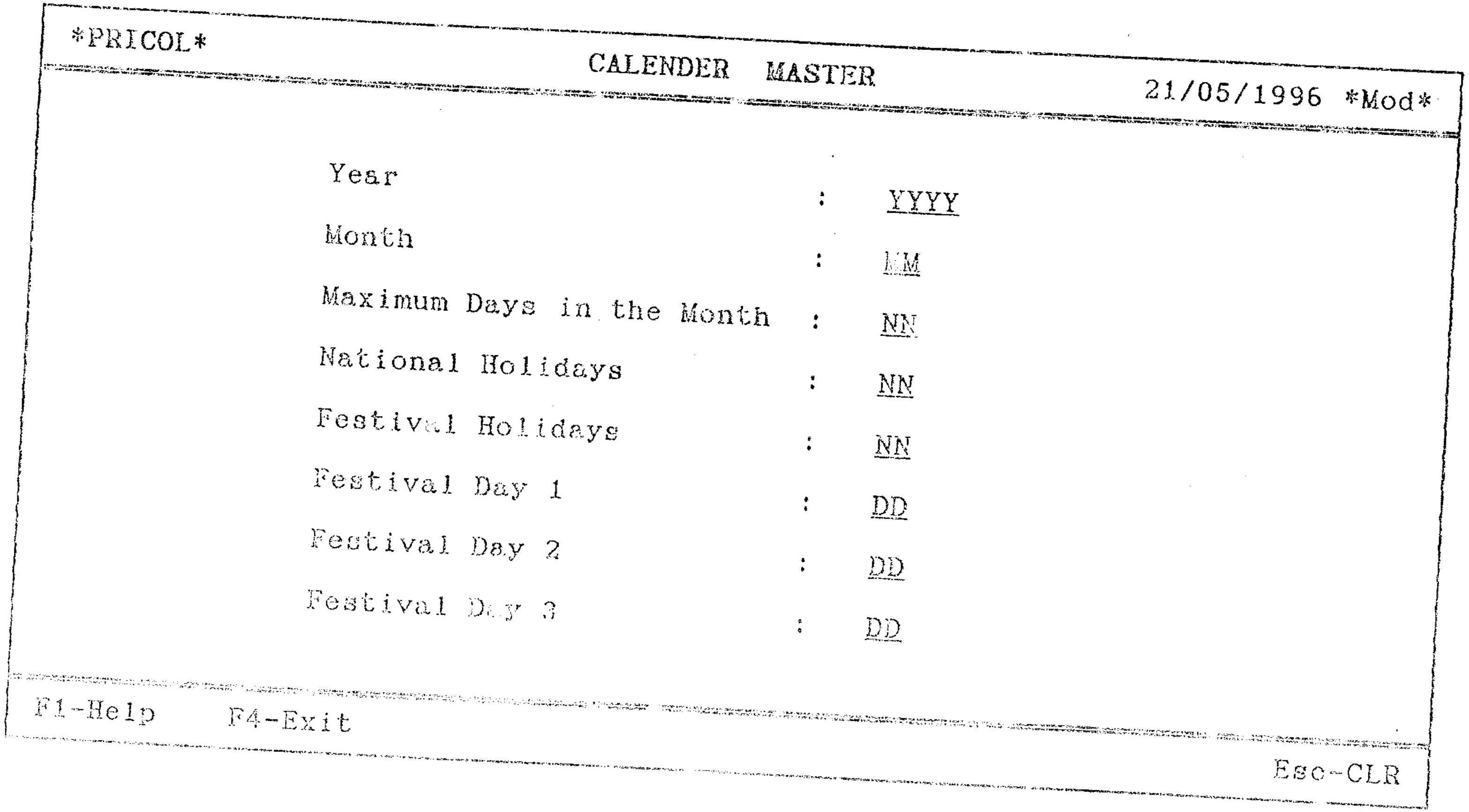


```
*PRICOL*                PAY  STRUCTURE  MASTER                21/05/1996 *Mod*

Category Code          : NN
Grade Code             : XXXXX

Minimum Basic          : NNNN.NN      Attendance Bonus          : NNNN.NN
Maximum Basic          : NNNN.NN      Bonus Amount              : NNNN.NN
Increment Amount       :  NNN.NN      Shift Allowance           :  NNN.NN

HRA                    : NNNN.NN      Medical Allowance         : NNNN.NN
CCA                    : NNNN.NN      Educational Allowance     : NNNN.NN
Vehicle Allowance      : NNNN.NN      Leave Travel Allowance    : NNNN.NN

Super Annuation Amount : NNNN.NN      Eligible Leave Days       : NN.NN
Car Allowance          : NNNN.NN

F1-Help    F4-Exit
                                                                    Esc-CLR
```

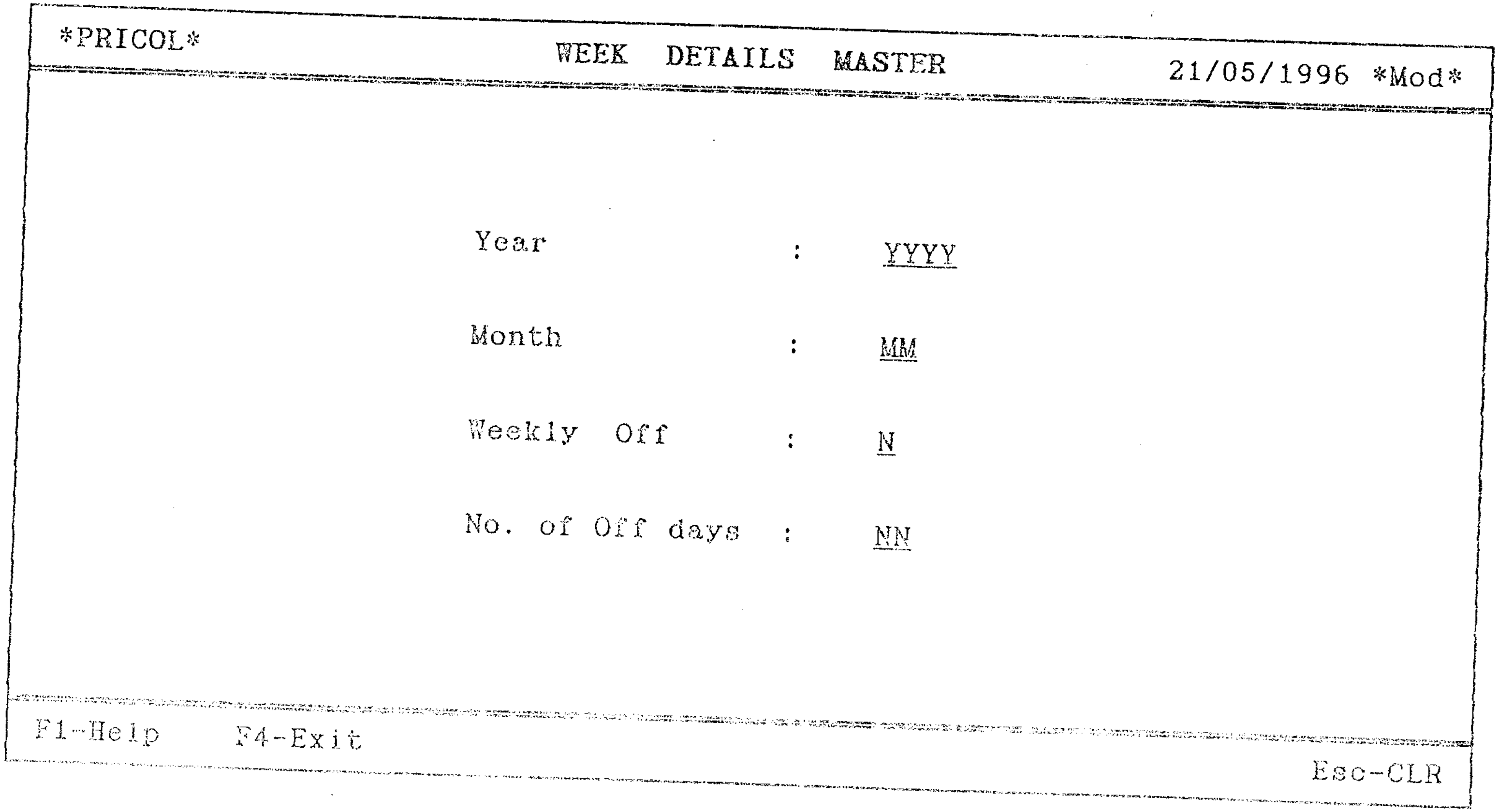


```
*PRICOL*
                         PAY  POINT  MASTER                      21/05/1996 *Mod*

                Pay Point Code    :  XX

                Description       :  XXXXXXXXXXXXXXXXXXXX

F1-Help   F4-Exit
                                                                           Esc-CLR
```

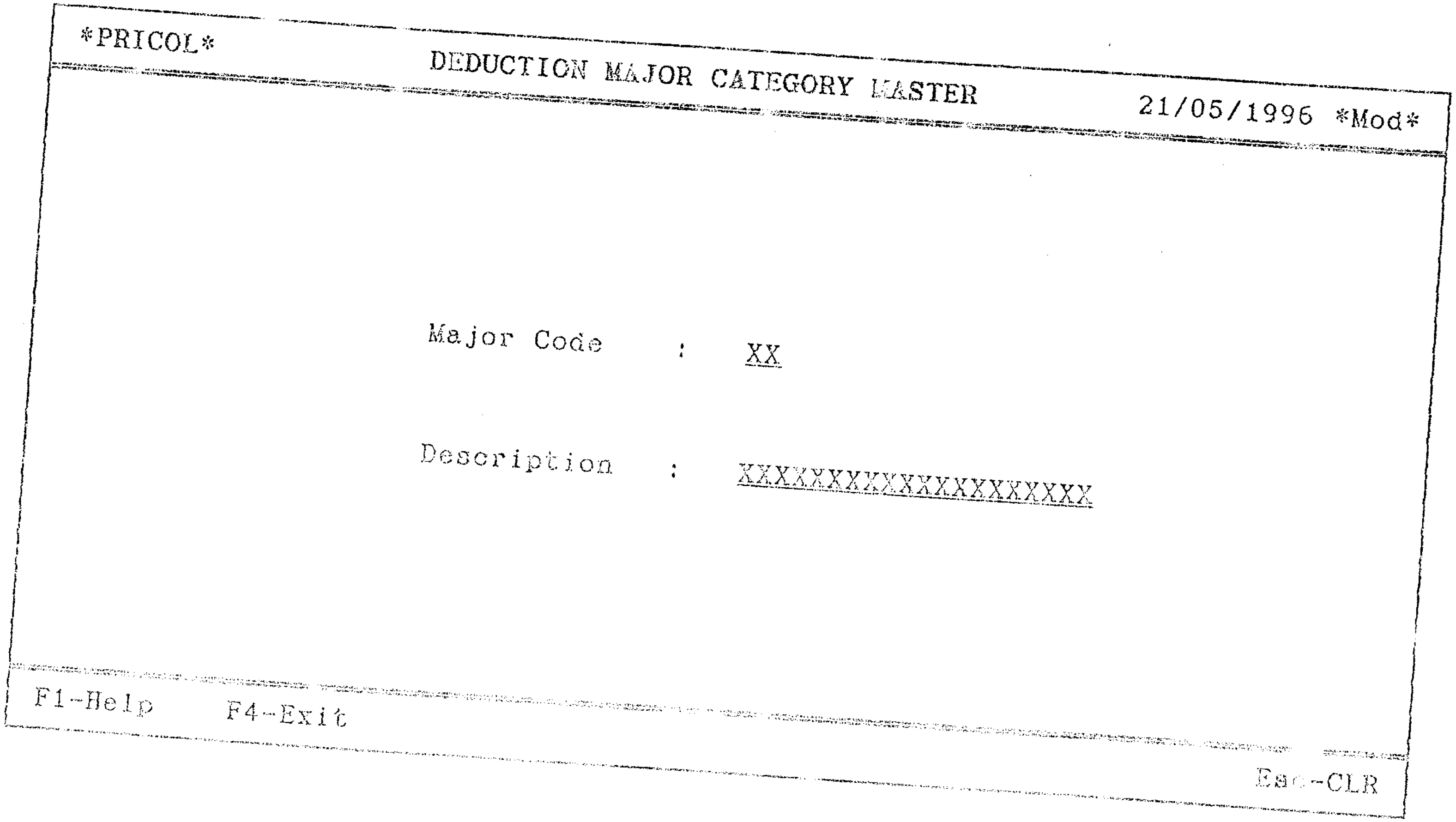




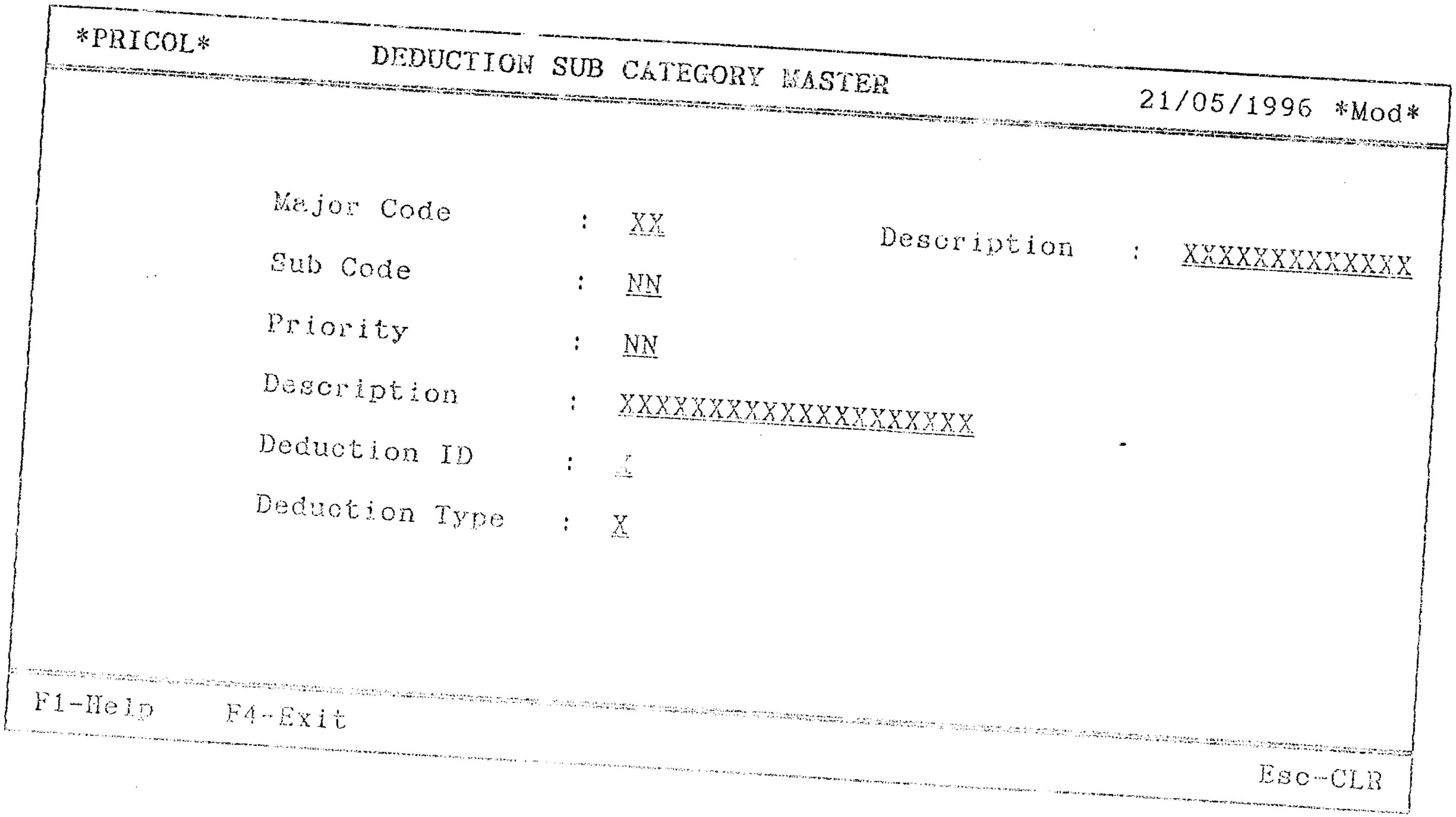
```
*PRICOL*
                         CALENDER  MASTER                    21/05/1996 *Mod*

          Year                              :   YYYY
          Month                             :   MM
          Maximum Days in the Month         :   NN
          National Holidays                 :   NN
          Festival Holidays                 :   NN
          Festival Day 1                    :   DD
          Festival Day 2                    :   DD
          Festival Day 3                    :   DD

F1-Help   F4-Exit
                                                                      Esc-CLR
```



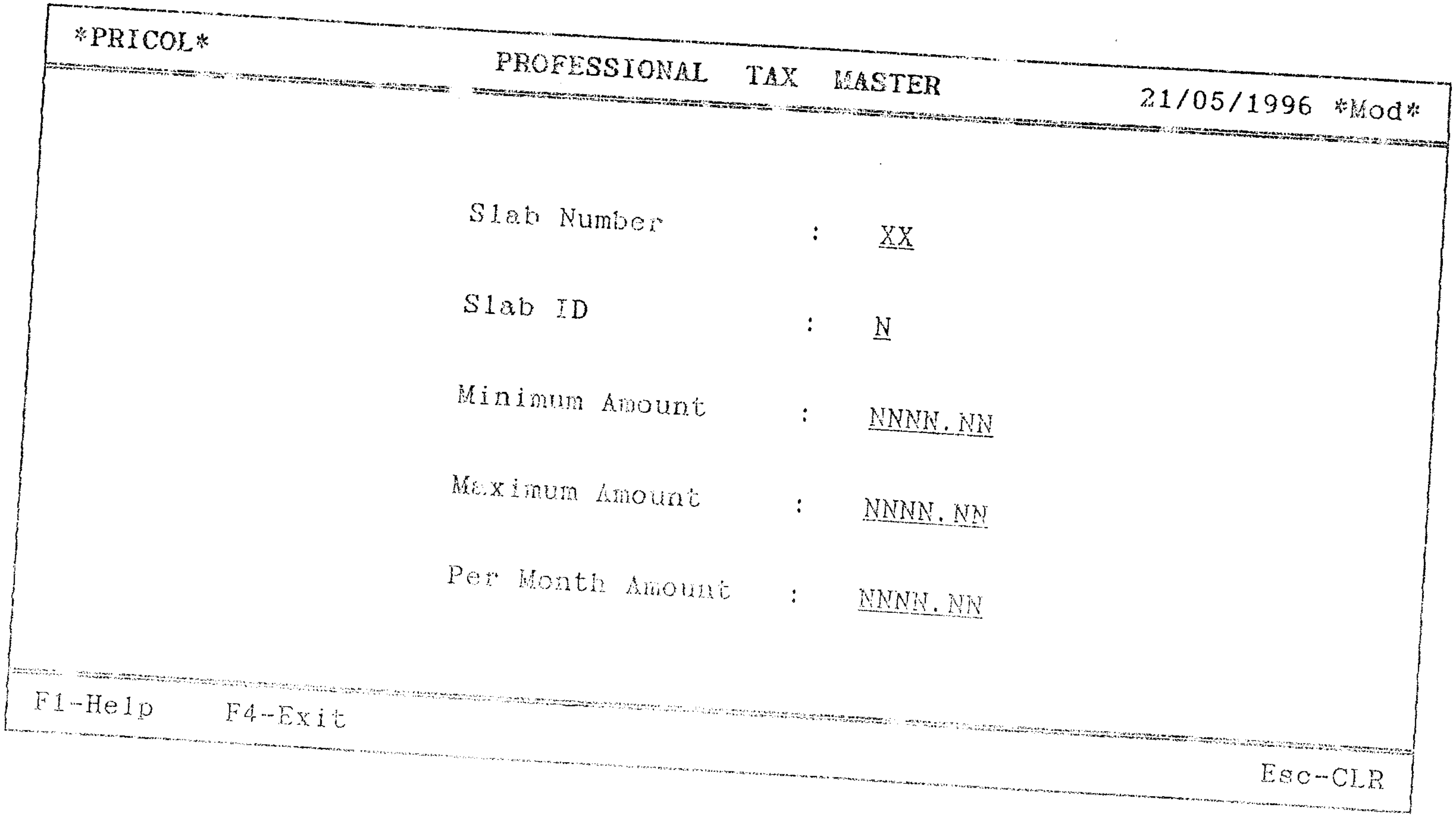
```
*PRICOL*                 WEEK  DETAILS  MASTER             21/05/1996 *Mod*

                    Year                :    YYYY

                    Month               :    MM

                    Weekly  Off         :    N

                    No. of Off days     :    NN

F1-Help    F4-Exit
                                                                    Esc-CLR
```



```
*PRICOL*
                    DEDUCTION MAJOR CATEGORY MASTER          21/05/1996 *Mod*

                    Major Code     :   XX

                    Description    :   XXXXXXXXXXXXXXXXXXXX

F1-Help    F4-Exit
                                                                      Esc-CLR
```



```
*PRICOL*          DEDUCTION SUB CATEGORY MASTER
                                                        21/05/1996 *Mod*

        Major Code       :  XX              Description  :  XXXXXXXXXXXXX
        Sub Code         :  NN
        Priority         :  NN
        Description      :  XXXXXXXXXXXXXXXXXXXX
        Deduction ID     :  X
        Deduction Type   :  X

F1-Help    F4-Exit
                                                                   Esc-CLR
```



PRICOL

PROFESSIONAL TAX MASTER

21/05/1996 *Mod*

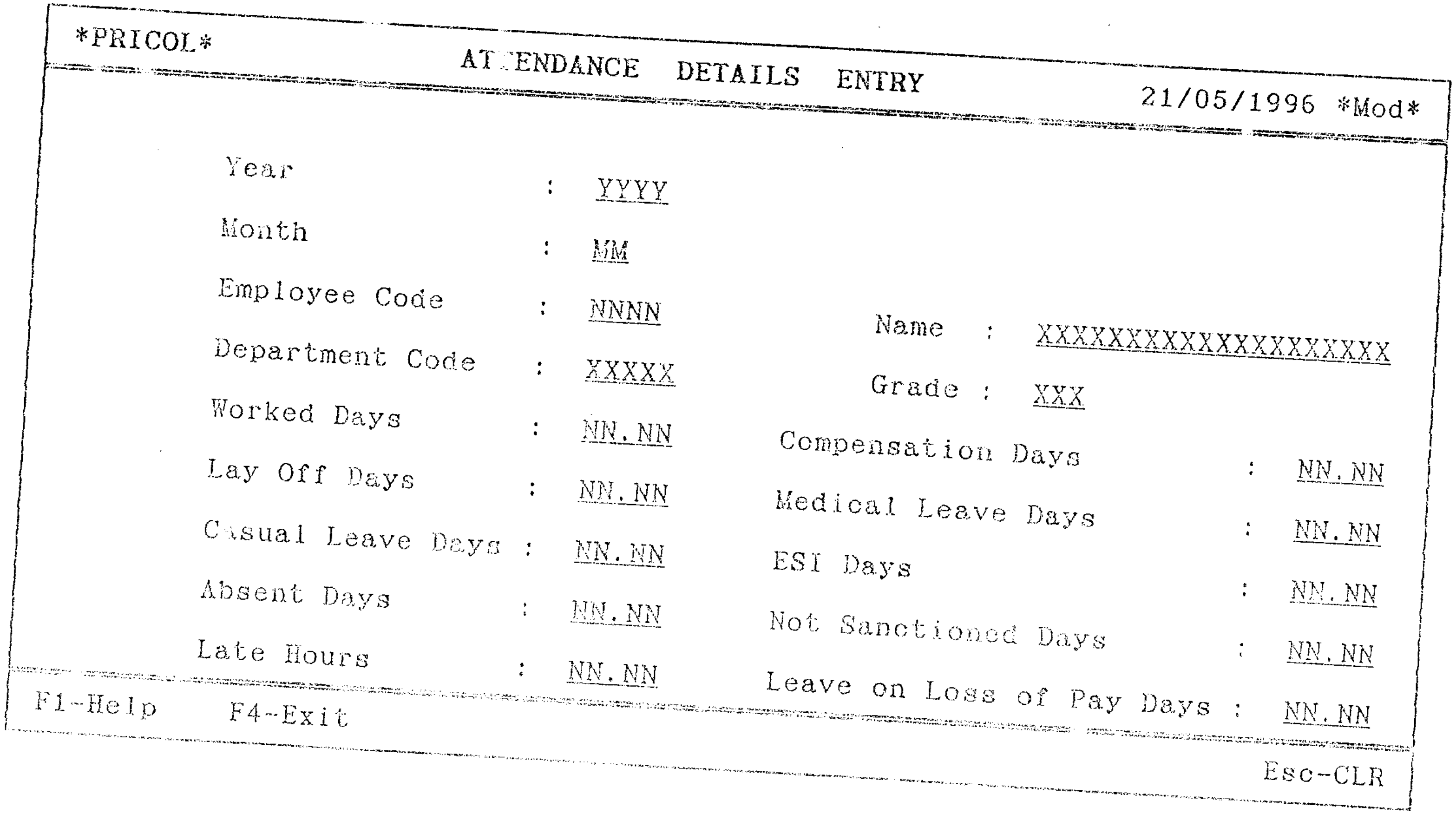
Slab Number : XX

Slab ID : N

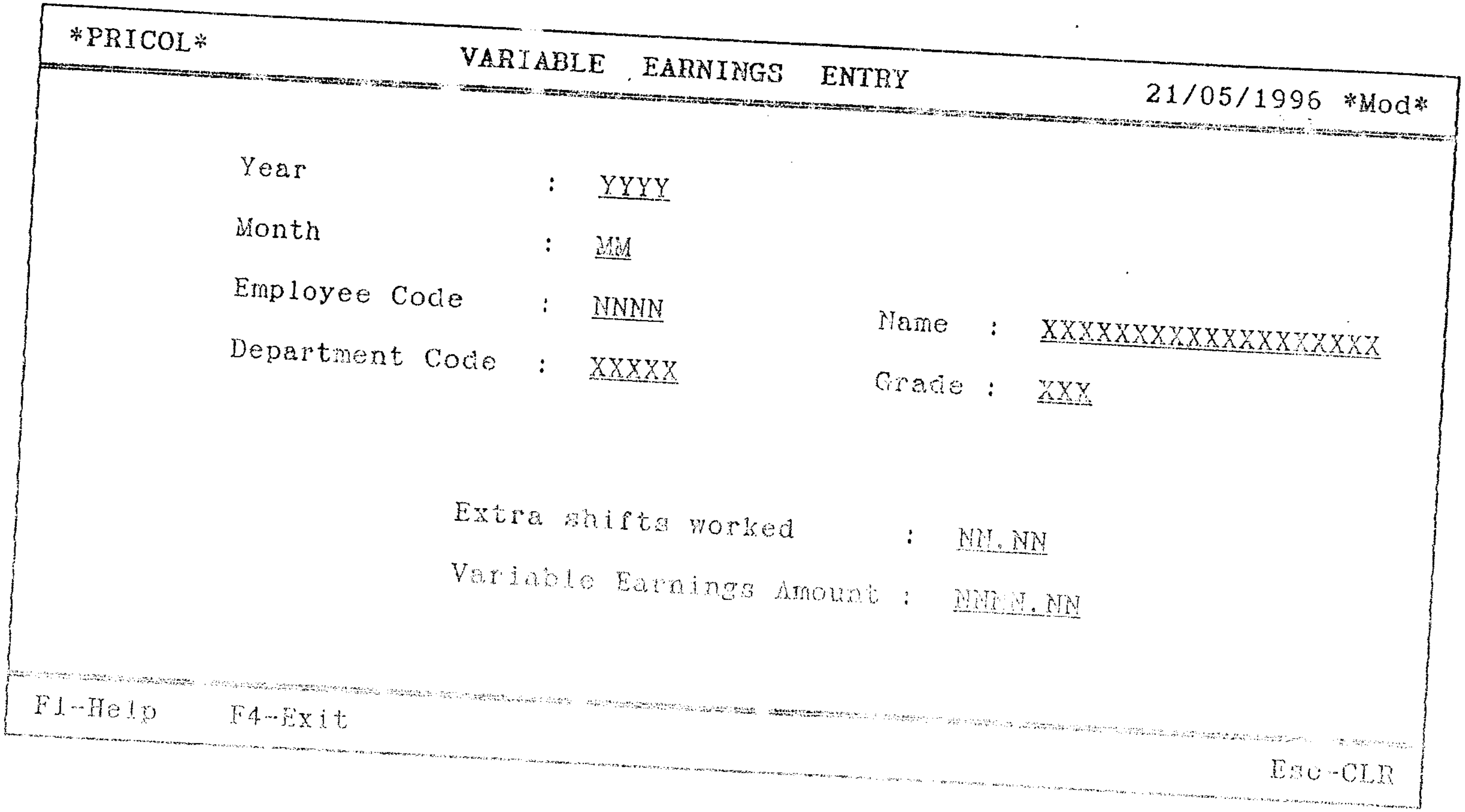
Minimum Amount : NNNN.NN

Maximum Amount : NNNN.NN

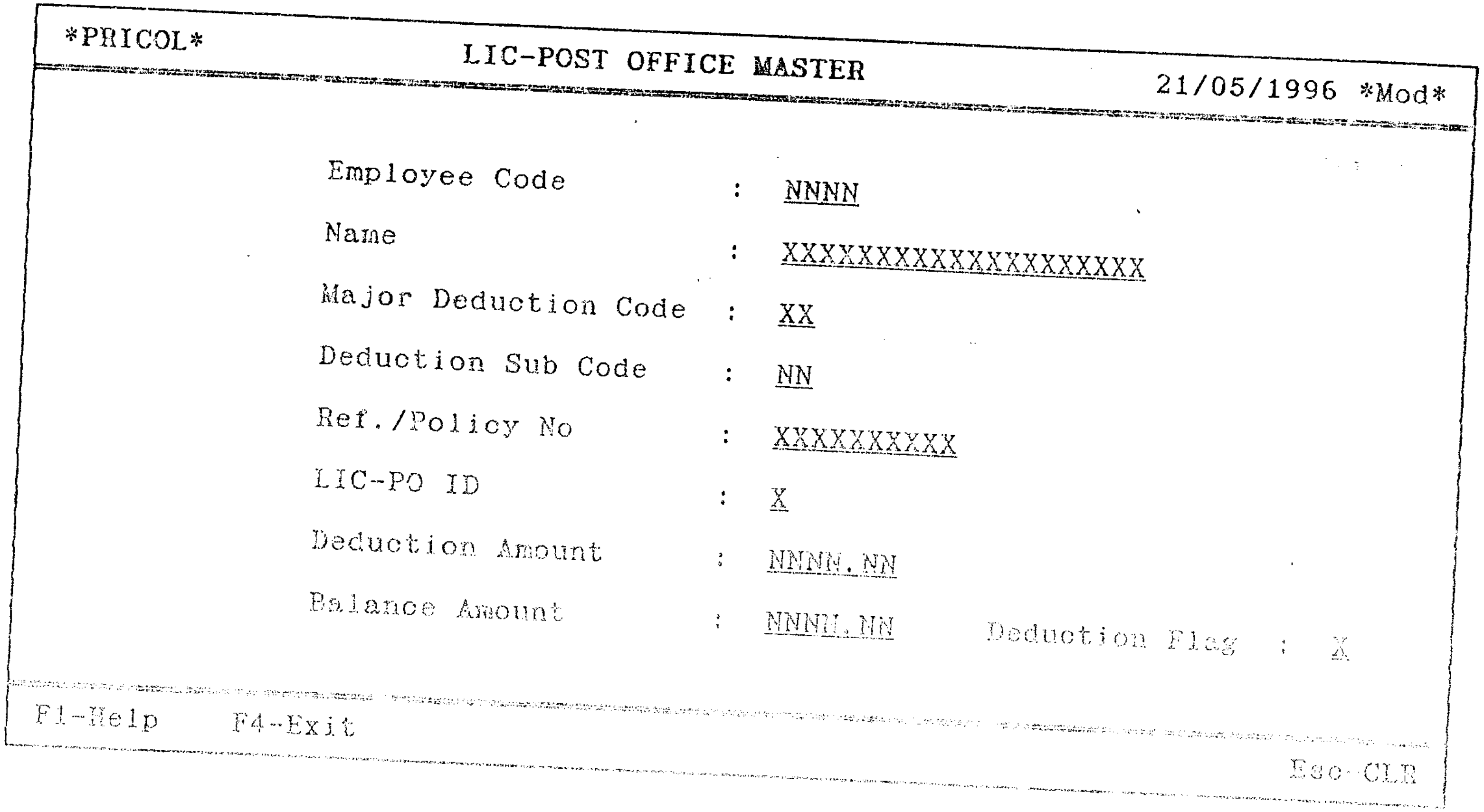
Per Month Amount : NNNN.NN



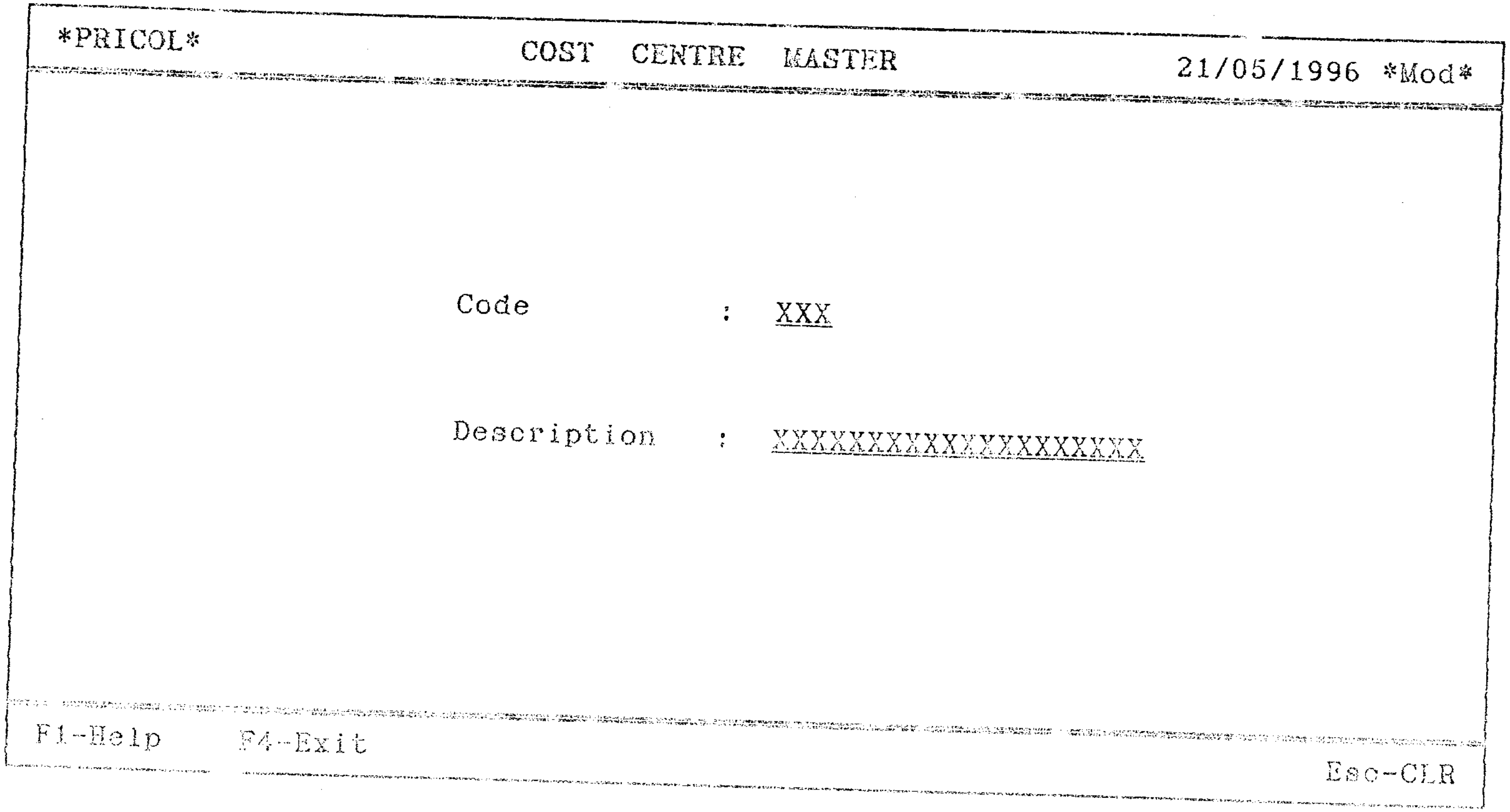
```
*PRICOL*
                     ATTENDANCE  DETAILS  ENTRY                    21/05/1996 *Mod*

      Year                  :  YYYY
      Month                 :  MM
      Employee Code         :  NNNN          Name  :  XXXXXXXXXXXXXXXXXXXX
      Department Code       :  XXXXX         Grade :  XXX
      Worked Days           :  NN.NN     Compensation Days            :  NN.NN
      Lay Off Days          :  NN.NN     Medical Leave Days           :  NN.NN
      Casual Leave Days     :  NN.NN     ESI Days                     :  NN.NN
      Absent Days           :  NN.NN     Not Sanctioned Days          :  NN.NN
      Late Hours            :  NN.NN     Leave on Loss of Pay Days    :  NN.NN
F1-Help    F4-Exit
                                                                         Esc-CLR
```



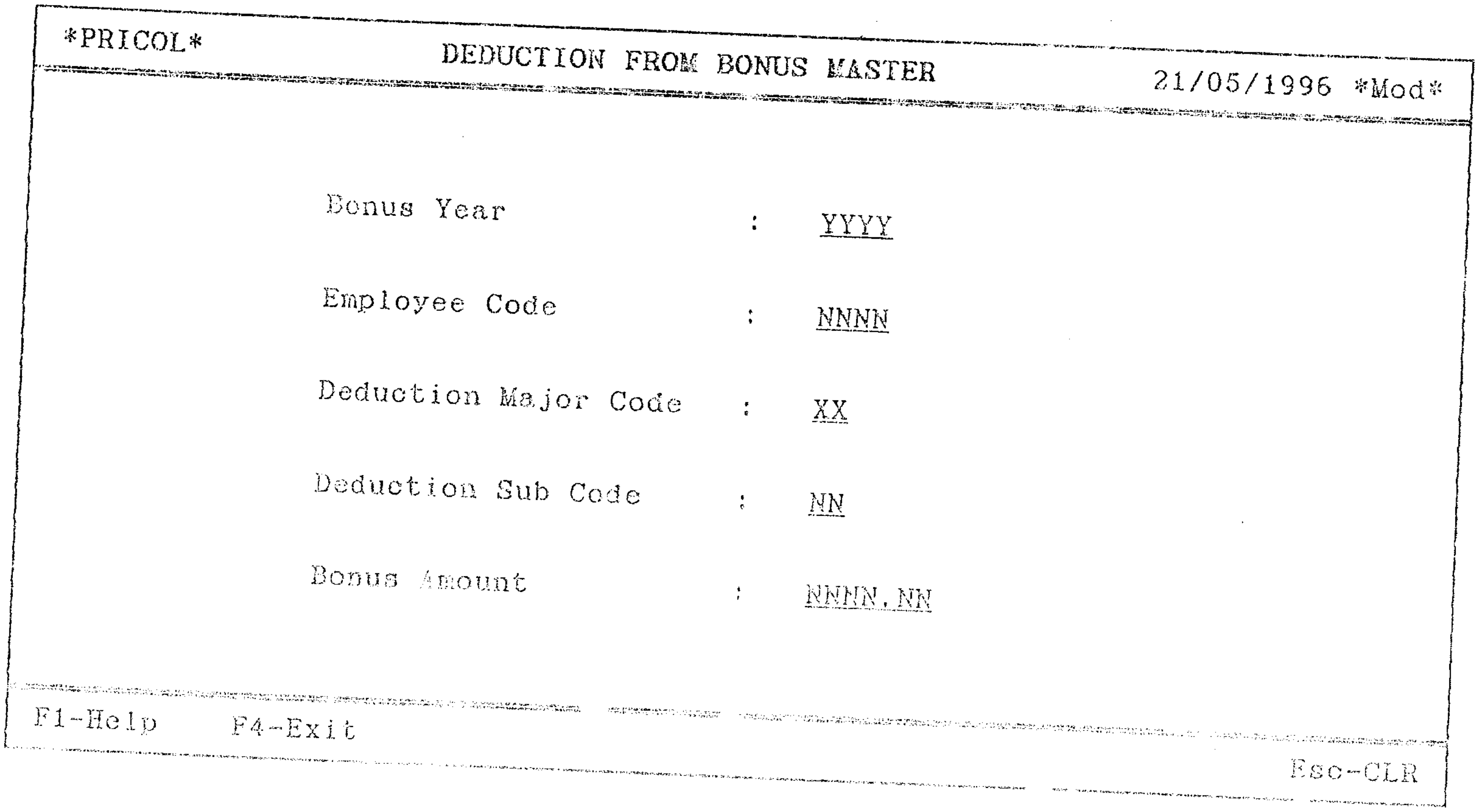
```
*PRICOL*
                    VARIABLE  EARNINGS  ENTRY                  21/05/1996 *Mod*

          Year              :  YYYY
          Month             :  MM
          Employee Code     :  NNNN             Name   :  XXXXXXXXXXXXXXXXXXX
          Department Code   :  XXXXX            Grade  :  XXX


                  Extra shifts worked        :  NN.NN
                  Variable Earnings Amount   :  NNNN.NN

F1-Help    F4-Exit
                                                                        Esc-CLR
```



```
*PRICOL*                 LIC-POST OFFICE MASTER                  21/05/1996 *Mod*

             Employee Code           :  NNNN
             Name                    :  XXXXXXXXXXXXXXXXXXXX
             Major Deduction Code    :  XX
             Deduction Sub Code      :  NN
             Ref./Policy No          :  XXXXXXXXXX
             LIC-PO ID               :  X
             Deduction Amount        :  NNNN.NN
             Balance Amount          :  NNNN.NN      Deduction Flag  :  X

F1-Help    F4-Exit
                                                                          Esc-CLR
```



```
*PRICOL*                 COST  CENTRE  MASTER                21/05/1996 *Mod*

                    Code           :  XXX

                    Description    :  XXXXXXXXXXXXXXXXXXXX

F1-Help  F4-Exit
                                                                    Esc-CLR
```



PRICOL DEDUCTION FROM BONUS MASTER 21/05/1996 *Mod*

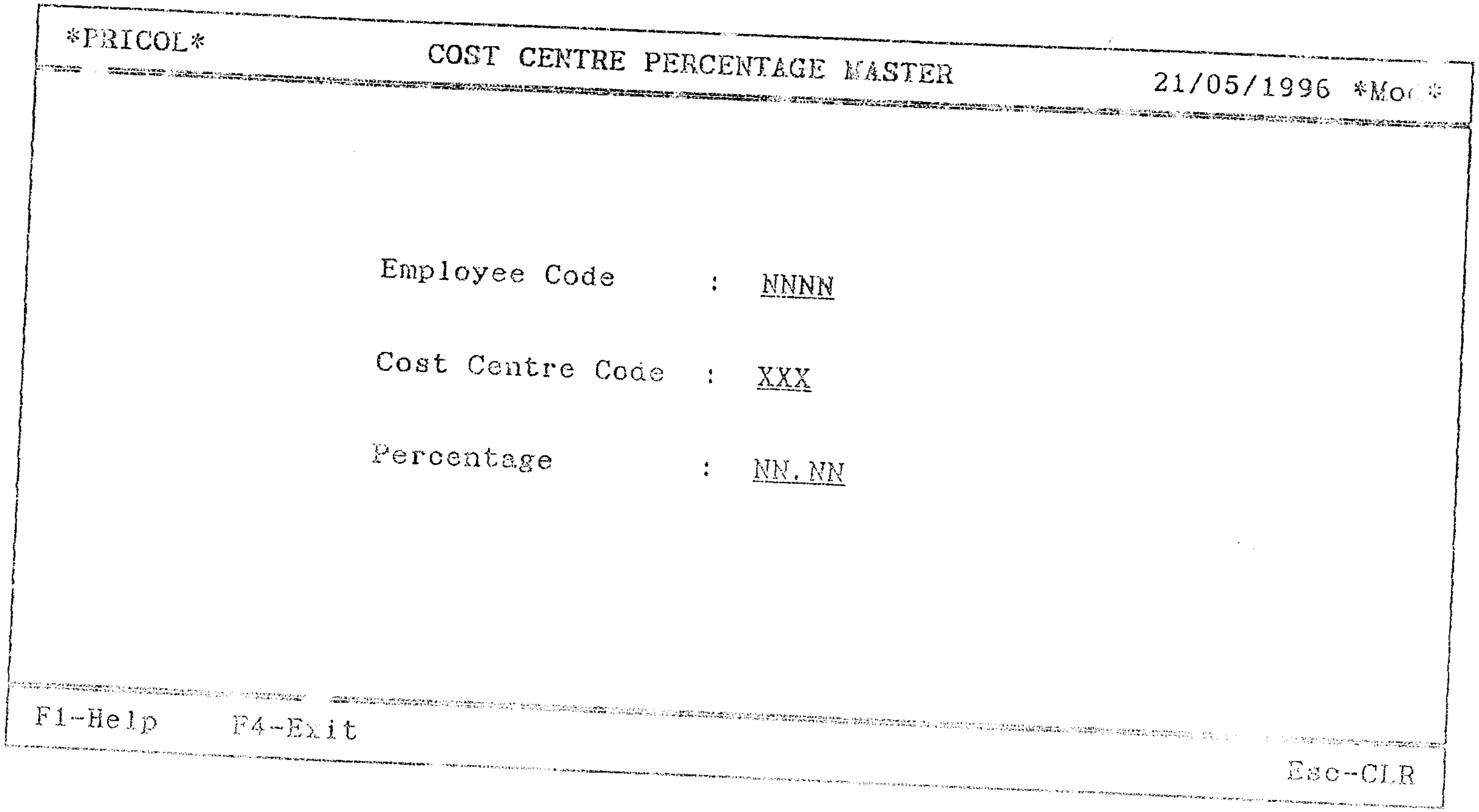
Bonus Year : YYYY

Employee Code : NNNN

Deduction Major Code : XX

Deduction Sub Code : NN

Bonus Amount : NNNN.NN

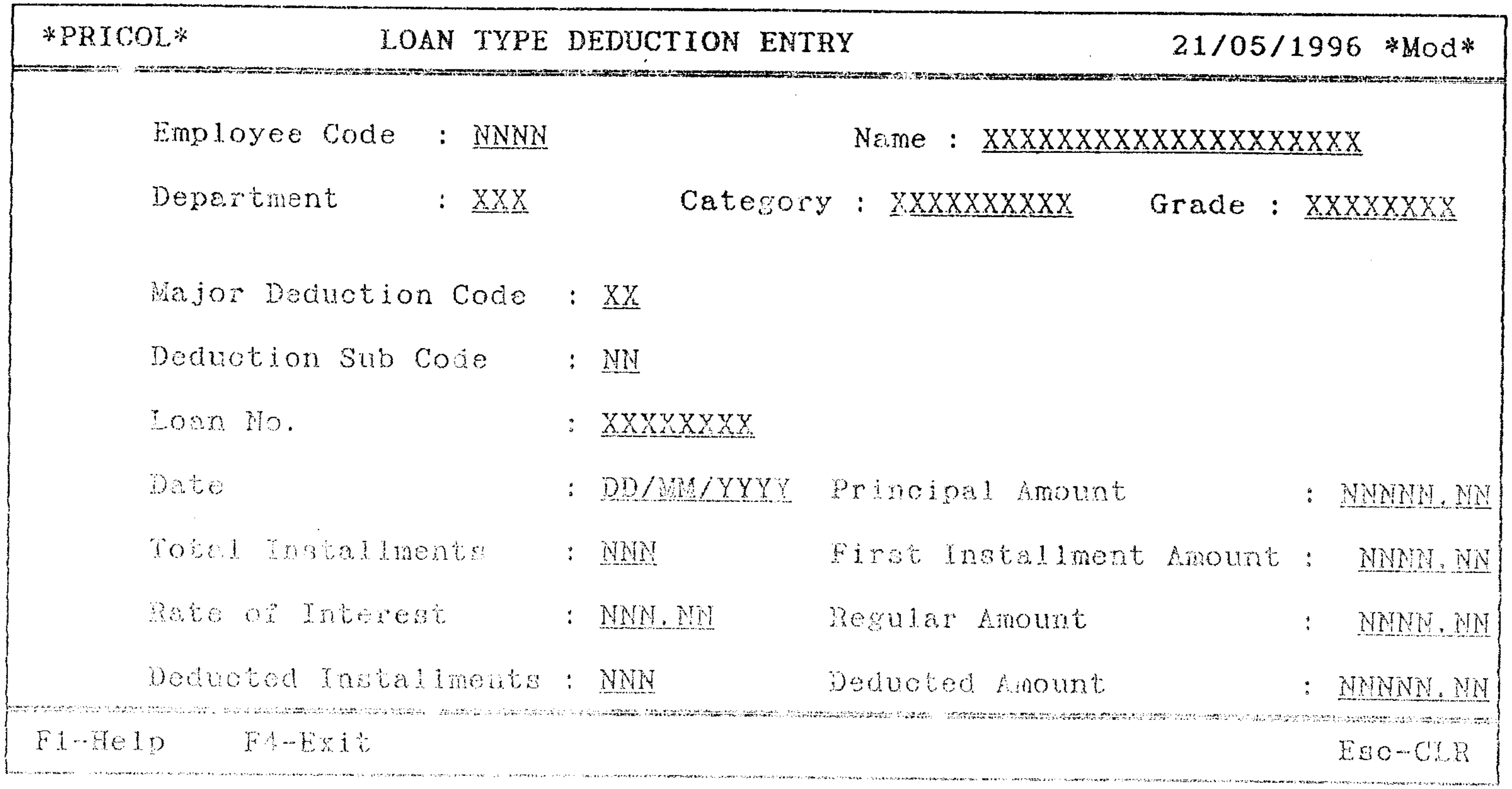


PRICOL COST CENTRE PERCENTAGE MASTER 21/05/1996 *Mo *

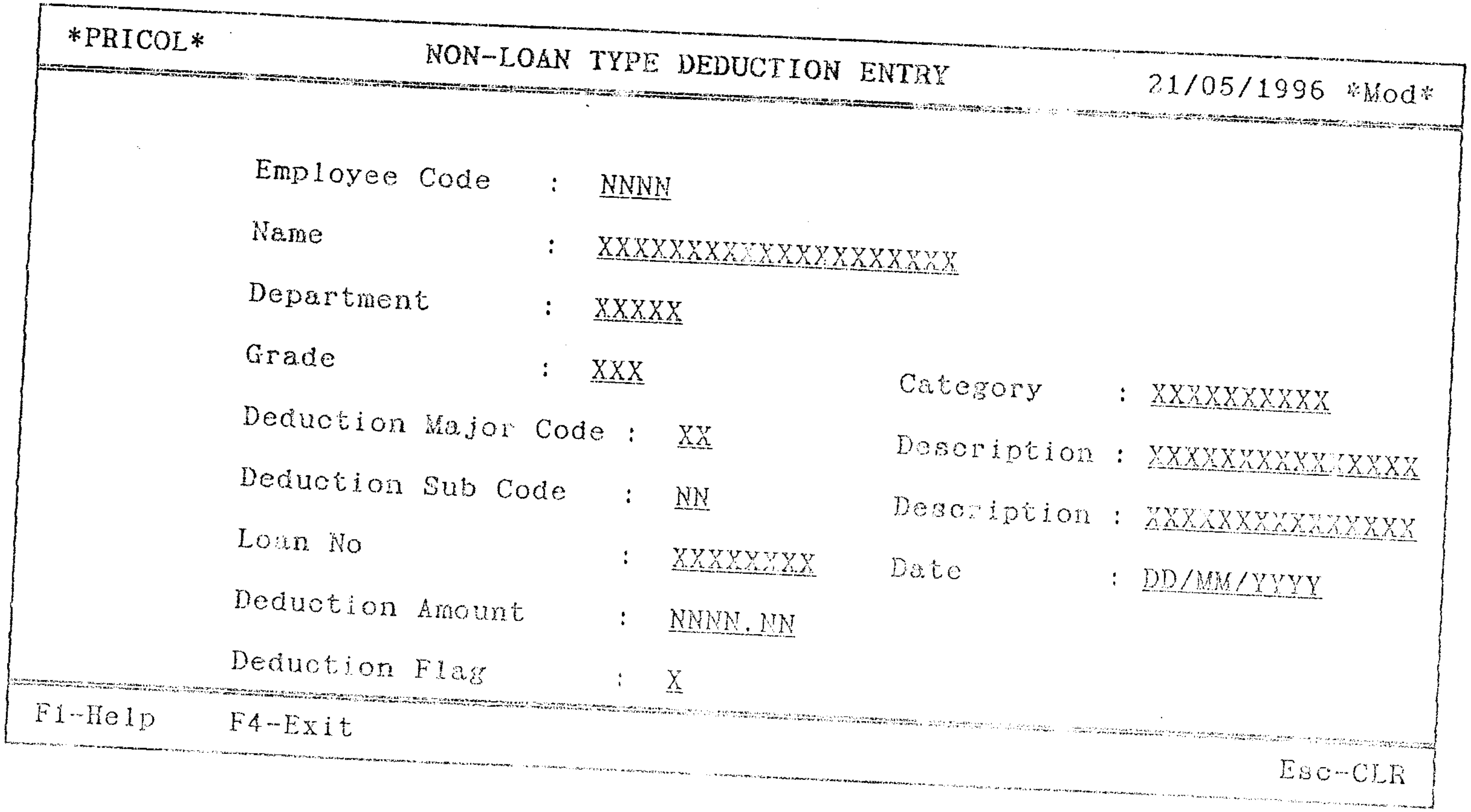
```
Employee Code      :  NNNN

Cost Centre Code   :  XXX

Percentage         :  NN.NN
```

F1-Help F4-Exit

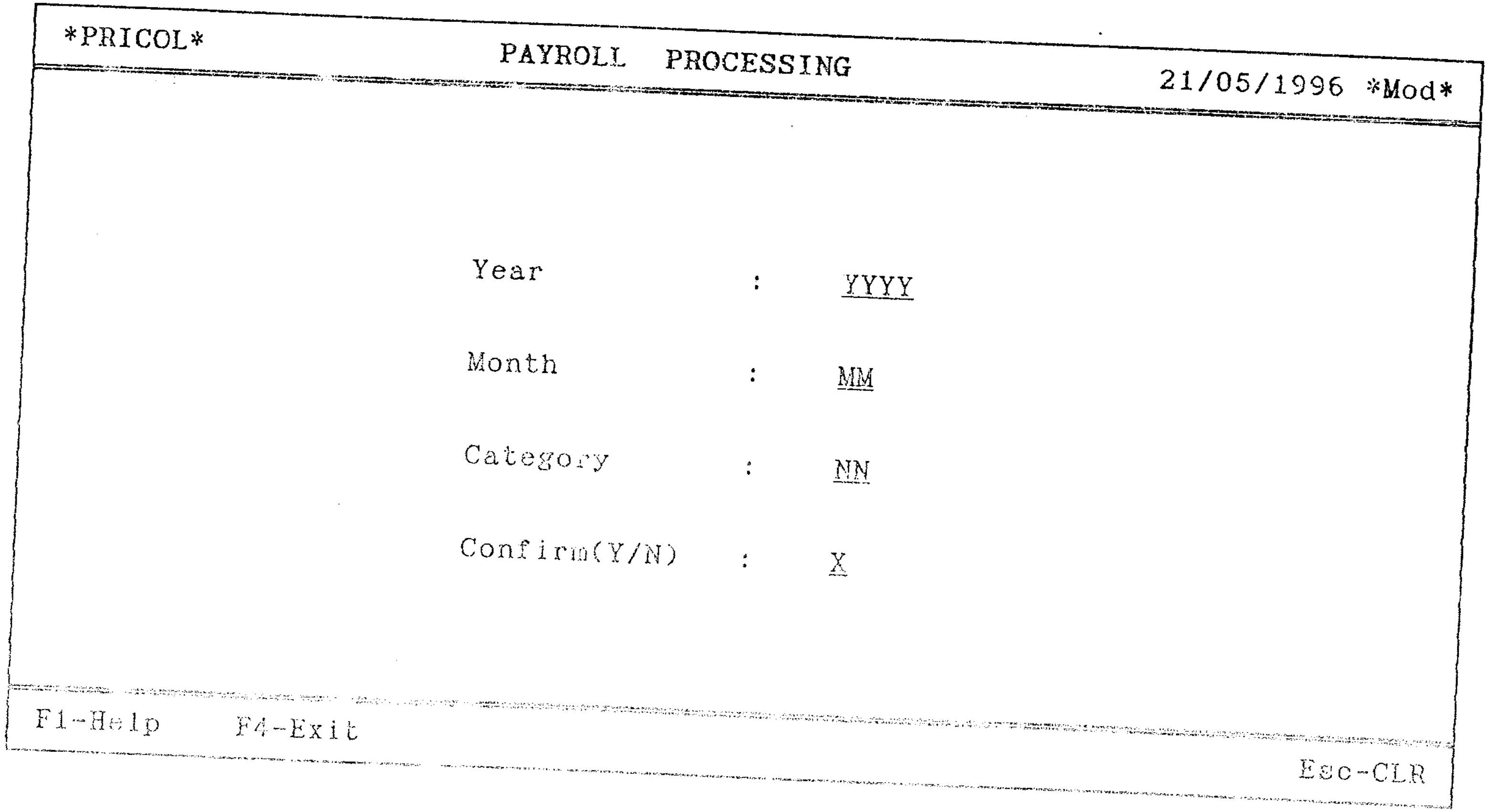
Esc-CLR



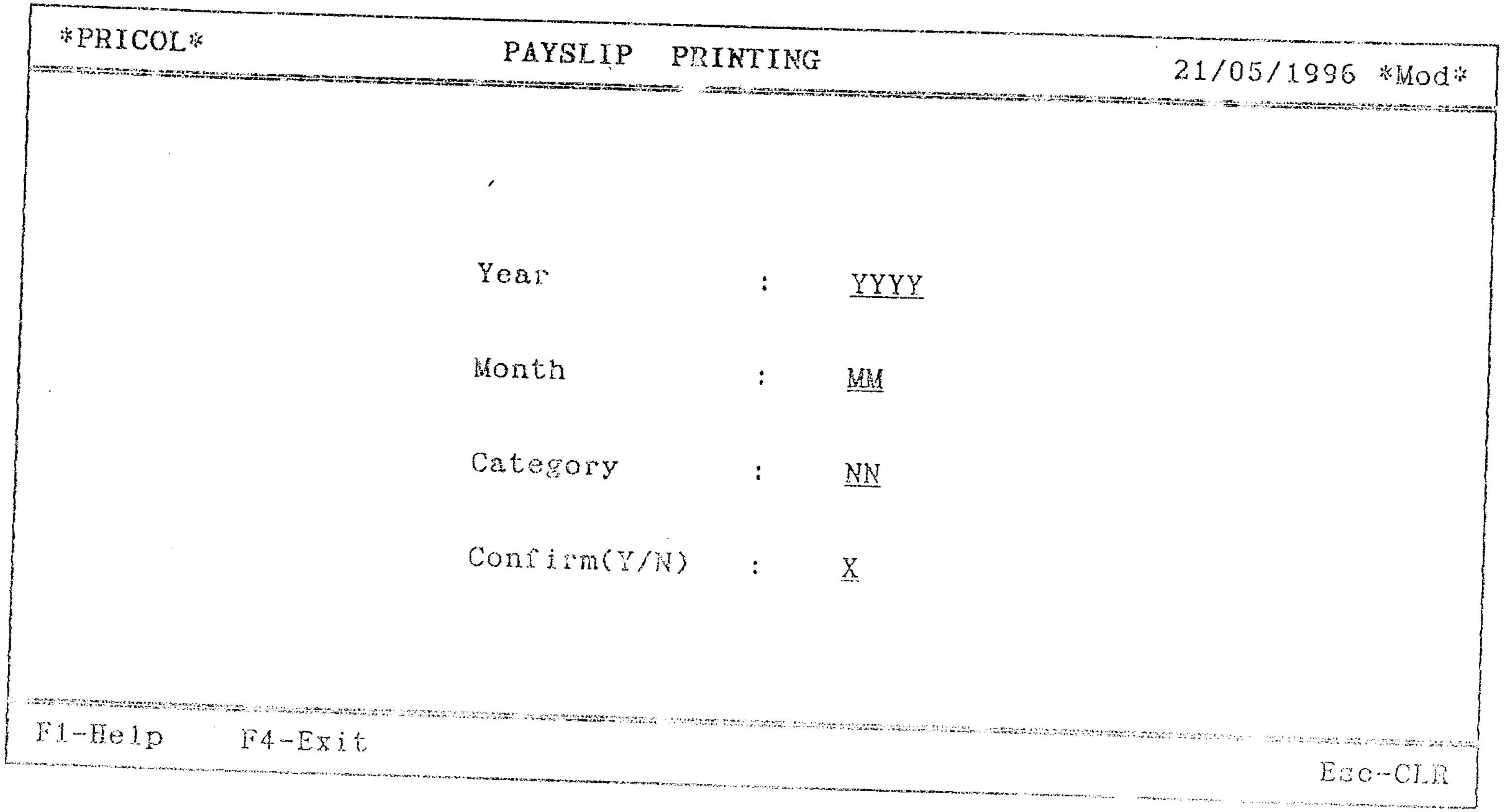
```
*PRICOL*            LOAN TYPE DEDUCTION ENTRY               21/05/1996 *Mod*

       Employee Code  : NNNN                Name : XXXXXXXXXXXXXXXXXXXX
       Department     : XXX        Category : XXXXXXXXXX    Grade : XXXXXXXX

       Major Deduction Code  : XX
       Deduction Sub Code    : NN
       Loan No.              : XXXXXXXX
       Date                  : DD/MM/YYYY  Principal Amount        : NNNNN.NN
       Total Installments    : NNN         First Installment Amount :  NNNN.NN
       Rate of Interest      : NNN.NN      Regular Amount          :  NNNN.NN
       Deducted Installments : NNN         Deducted Amount         : NNNNN.NN

 F1-Help   F4-Exit                                                   Esc-CLR
```



```
*PRICOL*                NON-LOAN TYPE DEDUCTION ENTRY                21/05/1996 *Mod*

      Employee Code        :  NNNN
      Name                 :  XXXXXXXXXXXXXXXXXXXX
      Department           :  XXXXX
      Grade                :  XXX               Category     : XXXXXXXXXX
      Deduction Major Code :   XX               Description  : XXXXXXXXXXXXXXX
      Deduction Sub Code   :   NN               Description  : XXXXXXXXXXXXXXX
      Loan No              :   XXXXXXXX         Date         : DD/MM/YYYY
      Deduction Amount     :   NNNN.NN
      Deduction Flag       :   X
F1-Help   F4-Exit
                                                                          Esc-CLR
```



```
*PRICOL*                  PAYROLL  PROCESSING              21/05/1996 *Mod*

                     Year               :    YYYY

                     Month              :    MM

                     Category           :    NN

                     Confirm(Y/N)       :    X

F1-Help    F4-Exit
                                                                   Esc-CLR
```



```
*PRICOL*                    PAYSLIP  PRINTING                 21/05/1996 *Mod*

                Year              :   YYYY

                Month             :   MM

                Category          :   NN

                Confirm(Y/N)      :   X

F1-Help    F4-Exit
                                                                       Esc-CLR
```

BIBLIOGRAPHY

1. Alex Berson and George Anderson

   – " SYBASE AND CLIENT/SERVER COMPUTING ",
   McGraw Hill Publication, 1995

2. Joe Salemi

   – " CLIENT/SERVER COMPUTING WITH SYBASE SQL SERVER ",
   BPB Publications, 1994

3. Open Client Release 10.0

   – " OPEN CLIENT DB-LIBRARY / C REFERENCE MANUAL ",
   Sybase Inc., 1993

4. Open Client Release 10.0

   – " SYBASE SQL SERVER REFERENCE MANUAL VOLUME I ",
   Sybase Inc., 1993

5. Open Client Release 10.0

   – " SYBASE SQL SERVER REFERENCE MANUAL VOLUME II ",
   Sybase Inc., 1993

6. Open Client Release 10.0

   – " TRANSACT SQL USER GUIDE ",
   Sybase Inc., 1993

7. Open Client Release 10.0

   – " DATA WORKBENCH USER'S GUIDE ",
   Sybase Inc., 1993

8. HP-UX 9000 SERIES MANUALS

9. Stephen Prata

- " ADVANCED UNIX - A PROGRAMMER'S GUIDE ",
  BPB Publications, 1986

10. Herbert Schildt

- " C : THE COMPLETE REFERENCE ",
  McGraw Hill Publication, 1987

11. C.J.Date

- " AN INTRODUCTION TO DATABASE SYSTEMS ",
  Narosa Publishing House, 1991

12. Roger S. Pressman

- " SOFTWARE ENGINEERING - A PRACTITIONER'S
  APPROACH ",
  McGraw Hill Publishers, 1992

13. Lee

- " INTRODUCING SYSTEMS ANALYSIS AND
  DESIGN ",
  Galgotia Booksource, 1982

'RICOL *

EMPLOYEE COST PARTICULARS

r : 1996

Month : Apr

Rep.Date :29/05/96 Page : 1

Category : STAFF

CODE	NAME	GROSS	PF	ESI	L.SAL	PERKS	GRAT	REWARD	CANT.	WELF.	TOT
1426	HARINARAYANAN C.	4860.00	328.65	-	300.73	578.00	38.17	153.34	-	57.68	8415
1428	PANDIAN.V	5330.00	354.90	-	220.77	972.00	41.73	161.24	-	63.44	9792.
1609	MUKESH PURI	6910.00	422.10	-	294.72	1623.30	46.48	193.23	-	84.69	13510
1612	RAGHURAM VITTAL R.	-	-	-	20.53	530.00	34.65		-	5.93	795.5
1613	RAJEEV MEHTA	5210.00	322.35	-	220.28	632.00	38.14	147.36	-	63.30	9047.7
1614	MOSES ISAIAH RAYMOND	5005.00	307.65	-	211.90	622.00	38.16	140.54		50.89	8709.87
1615	RAJAN ABRAHAM	4430.00	301.35	-	179.60	576.00	34.65	137.75		51.61	7499.3
1616	DEEPAK KUMAR MEGHLAN	4590.00	322.35	-	189.14	556.00	38.18	147.35		54.35	7972.61
1617	BIPLOB DEB	4050.00	298.20	-	161.27	530.00	34.65	136.52		46.40	8864.90
1618	PANKAJ TANEJA	5290.00	330.75	-	223.07	632.00	38.14	151.21		54.20	9174.02
1619	DILIP MAITHUL	5024.00	309.75	-	212.55	622.00	38.16	141.55		51.63	9739.01
1620	SHEKHAR SHRIKANT BHORKAR	4643.00	311.85	-	192.17	570.00	38.18	142.46		55.32	8061.13
1621	KULKARNI SHRINIVAS SHANKA	5043.00	311.85	-	213.33	622.00	38.16	142.46		51.27	8770.3
1622	SYED ILYAS HASSAN	5210.00	322.35	-	220.28	632.00	38.14	147.36		63.30	9047.72
1623	SUNDARARAJAN R	4043.00	311.85	-	165.82	496.00	38.19	142.46		47.56	7064.24
1624	PULIN BEHARI ROY	7060.00	396.90	-	310.59	1170.00	42.77	181.44	-	39.35	13063.17
1625	DIPANKAR DAS	4505.00	334.95	-	182.07	530.00	38.14	153.12		53.32	7791.07
1626	VIJAY M. BHAGAT	4760.00	318.15	-	197.25	578.00	38.17	145.44	-	56.59	8257.15
1627	KISHORE KANTI KAR	6758.00	379.05	-	297.02	1119.00	42.79	173.42	-	55.35	12505.90
1628	PRAKASH P.N.	4892.00	362.25	-	198.50	861.00	41.79	165.70	-	57.67	8959.37
1629	B.V.GOPAL	9219.00	505.05	-	402.11	2036.00	48.82	231.07		116.55	18199.35
1630	ARUN P.MEHTA	5210.00	322.35	-	220.28	632.00	38.14	147.35	-	55.30	9047.72
1631	SRINIVASA RAGHAVAN T.M.	6223.00	379.05	-	264.79	1029.00	42.91	173.42		56.09	11445.43
1632	PISAL T M	4160.00	283.50	-	168.92	560.00	33.32	129.60	-	48.54	7001.27
1633	MUTHAM PERUMAL R	4095.00	302.40	-	163.04	530.00	34.65	138.48		45.87	6933.00
634	RAVICHANDRAN K.S.	4643.00	311.85	-	192.17	570.00	38.18	142.46	-	55.32	8061.13
636	JATINDER PUNJ	12895.00	636.30	-	554.29	2840.00	-	291.12	-	189.78	17375.95

* End *

```
'RICOL *                      PAYROLL DEPARTMENTWISE EARNINGS ABSTRACT          Rep.Date : 29/05/96   Page : 1

.r : 1996                              Month : Apr                              Category : STAFF
--------------------------------------------------------------------------------------------------------------
:PARTMENT     BASIC PAY    FDA        HRA      LAY OFF   ATTND    NSHF    STR CON     GROSS     NETPAY      PFSAL
                STIPEND    VDA        CCA        WAGES   BONUS    ALWNS   N.CONF.     EARND
--------------------------------------------------------------------------------------------------------------
ELD STAFF     128558.00      -    44338.00         -       -        -          59   212843.00  186040.00  128558.00
                  -      16109.00      -                                        -
--------------------------------------------------------------------------------------------------------* End *
```

* PRICOL * DEPARTMENTWISE MAN HOURS LOST DETAILS Date : 20/05/96 Page : 1

Year : 1996 Month : Apr Category : STAFF

SNO	DEPT.	STR.	POSSIBLE_HRS	ACTUAL_HRS	MAN_HRS_LOST	PERC.
1	ADMIN	13	2600.00	298.50	2301.50	88.52
2	ATGI	2	400.00	41.00	359.00	89.75
3	ATINS	40	8000.00	930.50	7069.50	88.37
4	ATMTL	30	6000.00	700.00	5300.00	88.33
5	ATPPC	8	1600.00	191.00	1409.00	88.06
6	ATPRD	30	6000.00	715.50	5284.50	88.08
7	ATRND	51	10200.00	1109.00	9091.00	89.13
8	ATSTR	8	1600.00	194.00	1406.00	87.88
9	CPLAN	3	600.00	73.00	527.00	87.83
10	DEFEN	22	4400.00	474.50	3925.50	89.22
11	DRIVR	5	1000.00	113.50	886.50	88.65
12	ELINS	7	1400.00	159.50	1240.50	88.61
13	ELMTG	11	2200.00	259.50	1940.50	88.20
14	ELPRD	11	2200.00	220.00	1980.00	90.00
15	ELRND	21	4200.00	481.50	3718.50	88.54
16	FINCE	25	5000.00	561.50	4438.50	88.77
17	FPS	6	1200.00	139.50	1060.50	88.38
18	FSTAF	39	7800.00	953.00	6847.00	87.78
19	HRD	18	3600.00	402.50	3197.50	88.82
20	HTSHP	1	200.00	25.00	175.00	87.50
21	IGD	11	2200.00	268.50	1931.50	87.80
22	IMPRT	1	200.00	22.00	178.00	89.00
23	ISO	5	1000.00	118.00	882.00	88.20
24	MDOFF	1	200.00	21.00	179.00	89.50
25	MFENG	23	4600.00	510.00	4090.00	88.91
26	MIS	14	2800.00	283.50	2516.50	89.88
27	MSHOP	9	1800.00	190.00	1610.00	89.44

*** PRICOL * DEPARTMENTWISE MAN HOURS LOST DETAILS** Date : **20/05/96** Page : 2

Year : **1996** Month : **Apr** Category : **STAFF**

SNO	DEPT.	STR.	POSSIBLE_HRS	ACTUAL_HRS	MAN_HRS_LOST	PERC.
28	PMD	22	4400.00	504.00	3896.00	88.55
29	PRTNG	5	1000.00	112.00	888.00	88.80
30	PUMPS	40	8000.00	956.50	7043.50	88.04
31	SALES	31	6200.00	683.50	5516.50	88.98
32	SECTL	6	1200.00	139.00	1061.00	88.42
33	TROOM	32	6400.00	777.50	5622.50	87.85
34	VENDV	4	800.00	99.00	701.00	87.62
35	VPOFF	7	1400.00	160.00	1240.00	88.57
	Total		112400.00	12887.00	99513.00	3099.96

* End *

*** PRICOL *** **DEPARTMENTWISE STRENGTH DETAILS** Date : **21/05/96** Page : 1

Category : **STAFF**

SNO	DEPARTMENT	CONFIRMED	NON-CONFIRMED	TOTAL
1	ADMINISTRATION	11	2	13
2	AUTO - GI	2	-	2
3	AUTO - INSPECTION	34	6	40
4	AUTO - MATERIALS	25	5	30
5	AUTO - PPC	7	1	8
6	AUTO - PRODUCTION	25	5	30
7	AUTO - R&D	49	2	51
8	AUTO - STORES	8	-	8
9	CORPORATE PLANNING	3	-	3
10	DEFENCE	20	2	22
11	DRIVERS	5	-	5
12	ELECTRONICS-INSP	7	-	7
13	ELECTRONICS-MKTG	10	1	11
14	ELECTRONICS-PROD	11	-	11
15	ELECTRONICS - R&D	17	4	21
16	FINANCE	25	-	25
17	F.P.S	6	-	6
18	FIELD STAFF	39	-	39
19	H.R.D	16	2	18
20	H T S	1	-	1
21	I G D	9	2	11
22	IMPORTS	1	-	1
23	ISO CELL	5	-	5
24	MD OFFICE	1	-	1
25	MANUFACT. ENGG.	12	11	23
26	M.I.S	13	1	14
27	MACHINE SHOP	7	2	9

* PRICOL *　　ESI DEDUCTION DETAILS　　Date : 11/05/96　　Page

Year : 1996　　Month : Feb　　Category : OPERATOR

SNO	EMP NO	NAME	ESI NO	NO OF DAYS WORKED	GROSS EARNED	ESI AMOUN
1	2	VARADARAJAN S.K.	10484674	24.50	3151.27	47.2
2	1	LOGANATHAN V B	10489579	25.00	3242.62	48.6
		Total			6393.89	95.9

* En

*** PRICOL *** **GRADEWISE STRENGTH DETAILS** Date : **21/05/96** Page : 1

Category : **STAFF**

SNO	GRADE	CONFIRMED	NON-CONFIRMED	TOTAL
1	01A	14	-	14
2	01AD	1	-	1
3	01B	9	-	9
4	02A	101	-	101
5	02AC	4	-	4
6	02AD	2	-	2
7	02B	20	-	20
8	03A	41	-	41
9	03AA	1	-	1
10	03AB	1	-	1
11	03AC	2	-	2
12	03AD	4	-	4
13	03B	47	-	47
14	03BC	3	-	3
15	03BD	4	-	4
16	03BE	1	-	1
17	04A	39	-	39
18	04AB	1	-	1
19	04AC	2	-	2
20	04B	48	-	48
21	04BA	1	-	1
22	04BB	1	-	1
23	04BC	1	-	1
24	04BD	1	-	1
25	05A	27	-	27
26	05AD	1	-	1
27	05B	26	-	26

* PRICOL * DEDUCTIONS LIST Date : 11/05/96 Page : 1

Year : 1996 Month : Feb Category : OPERATOR

Deduction : Professional Tax

SNO	EMP.CODE	NAME	AMOUNT
1	1	LOGANATHAN V B	10.00
2	2	VARADARAJAN S.K.	10.00
3	1583	THANGARAJ R.	18.00
4	1585	GURUSAMY P.	18.00
5	1587	DHANALAKSHMI V.	18.00
6	1591	PRADEEP KUMAR P.	18.00
7	1592	SIVARAJ R.	18.00
8	1593	NANDAKUMAR C S.	18.00
9	1598	MURTHY P.	18.00
10	1599	JAYACHITRA R.	18.00
11	1600	NIRMALA S.	18.00
12	1602	RAJESWARI R.	18.00
13	1603	GOMATHI R.	18.00
14	1606	SENTHILKUMAR .R	18.00
15	1607	SARAVANAKUMAR .R	18.00
16	1608	MAHENDRAN .A	18.00
	TOTAL		272.00

* End *

*** PRICOL *   EPF, FPF DEDUCTION DETAILS**   Date : 11/05/96   Page

Year : **1996**   Month : **Feb**   Category : **STAFF**

SNO	EMP.CODE	NAME	PF. NO	PF	DEDUCTION		
					EMPLOYEE EPF (10%)	EPF (1.67%)	EMPLOYE FPF (8.33
		Balance brought Forward		64882	6486	6486	
24	1183	VENKATESH D.	2108	2748	275	275	
25	1188	PUGAZENTHI R .	2181	2825	282	282	
26	1191	RAMANATHAN S.	2189	2840	284	284	
27	1190	RAVICHANDRAN S.	2190	2405	240	240	
28	1668	KATHIRESH. M	2262	3210	321	321	
29	1716	RADHAKRISHNAN .R	2310	2260	226	226	
30	1722	VARATHARAJAPERUMAL .	2316	2530	253	253	
		TOTAL		83700	8367	8367	

* End

* PRICOL *   SUPER ANNUATION EARNINGS DETAILS   Rep.Date : 9/05/96   Page

Year : 1996   Month : Apr   Category : STAFF

SNO	CODE	NAME	AMOUNT
1	1609	MUKESH PURI	402.00
2	1629	B.V.GOPAL	467.00
3	1636	JATINDER PUNJ	758.00
4	1638	UNNI KRISHNAN K.	402.00
5	1645	MANMOHAN PILLAY K.	726.00
	Total		2755.00

* End *