



D Y PATIL
— SCHOOL OF —
POLYTECHNIC
NAVI MUMBAI

An Industrial Training (22057)

Report

(15th June, 2020 - 30th July, 2020)

Submitted by

Shreyas Varadkar

Under the Guidance of

Prof. Sunny Rane/Prof. Alok Gupta
MICA Labs, CBD-BELAPUR

Prof. Poonam Jadhav
Faculty Mentor

Department of Computer Engineering
DR. D. Y. PATIL POLYTECHNIC,
NERUL, NAVI MUMBAI
Academic Year 2020-2021

ACKNOWLEDGEMENT

I take this opportunity to express my profound gratitude and deep regard to **Prof. Sunny Rane and Prof. Alok Gupta**, for their explanatory guidance, monitoring and constant encouragement throughout the course of the internship. The blessings and the guidance given by them will go a long way in the years to come.

I would also like to express my deep sense of gratitude to my faculty mentor **Prof. Poonam Jadhav** for her valuable feedback and guidance which helped me in completing this internship.

I am thankful to **Prof. Umesh Patil** Head, Department of Computer Engineering for facilitating us with this six weeks internship program to get more exposure and practical knowledge on the topic of interest.

Shreyas Varadkar

INDEX

| Chapter No. | Title | Page No. |
|--------------------|---|-----------------|
| 1 | Objectives of Internship | 1 |
| 2 | Introduction of Industry | 2 |
| 3 | Introduction to Python Programming | 3-10 |
| 4 | Introduction to MySQL | 11-18 |
| 5 | Description of the Project | 20-29 |

1.0 Objectives of Internship

The purpose of Industrial Training is to expose students to real work of environment experience and at the same time, to gain the knowledge through hands on observation and job execution. From the industrial training, the students will also develop skills in work ethics, communication, management and others. Moreover, this practical training program allows students to relate theoretical knowledge with its application in the manufacturing industry.

The objectives of industrial training are:

- To provide students the opportunity to test their interest in a particular career before permanent commitments are made.
- To develop skills in the application of theory to practical work situations.
- To develop skills and techniques directly applicable to their careers.
- Internships will increase a student's sense of responsibility and good work habits.
- To expose students to real work environment experience gain knowledge in writing report in technical works/projects.
- Internship students will have higher levels of academic performance.
- Internship programs will increase student earning potential upon graduation.
- To build the strength, teamwork spirit and self-confidence in students life.
- To enhance the ability to improve students creativity skills and sharing ideas.
- To build a good communication skill with group of workers and learn to learn proper behaviour of corporate life in industrial sector.
- The student will be able instilled with good moral values such as responsibility commitment and trustworthy during their training.

2.0 Introduction to Industry

MICA Labs located in CBD Belapur is a leading solution provider of marine and industrial automation, software development and Engineering solution since 2003. We provide solution to the clients based on embedded system and do projects in microcontroller based system, microprocessor and on pic controller. Mainly we focus on C programming language to execute the projects.

MICA Labs provides platform to employees and staff in the field of embedded, research and development.

3.0 Introduction to Python Programming

- Python is an interpreted, high-level and general-purpose programming language. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically-typed and garbage-collected. It supports including structured (particularly, procedural), object oriented and functional programming.
- Python was first released in 1991. by Guido van Rossum as a successor to the ABC programming language.
- Python 2.0. released in 2000, introduced new features, such as list comprehensions, and a garbage collection system with reference counting, and was discontinued with version 2.7 in 2020. Python 3.0. released in 2008, was a major revision of the language that is not completely backward-compatible and much Python 2 code does not run unmodified on Python 3. With Python 2's end-of-life (and pip having dropped support in 2021), only Python 3.6.x and later are supported, with older versions still supporting eg.. Windows 7 (and old installers not restricted to 64-bit Windows).

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way

Python Syntax compared to other programming languages

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Python Comments

- Comments can be used to explain Python code.
- Comments can be used to make the code more readable.
- Comments can be used to prevent execution when testing code.

Python Data Types

- Built-in Data Types
- Getting Data Types
- Setting Data Types
- Setting specific Data Types

Python Numbers

- int (signed integers).
- long (long integers)

- float (floating point real values)
- complex (complex numbers.

Python Casting

- Specify a Variable Type - There may be times when you want to specify a type on to a variable. This can be done with casting. Python is an object-orientated language, and as such it uses classes to define data types, including its primitive types.

Python Strings

- Assign String to a Variable
- Multiline Strings
- Strings are Arrays
- Looping Through a String
- String Length
- Check String
- Check if NOT

Python Operators

- Arithmetic operators
- Assignment operators
- Comparison operators
- Logical operators

- Identity operators
- Membership operators
- Bitwise operators

Python Lists

- Lists are used to store multiple items in a single variable.
Lists are one of 4 built-in data types in Python used to store collections of data, the other 3 are tuple, set, and, all with different qualities and usage.
Lists are created using square brackets:

Python Tuples

- Tuples are used to store multiple items in a single variable. Tuple is one of 4 built-in data types in Python used to store collections of data, the other 3 are List, Set, and Dictionary, all with different qualities and usage.

Python Loops

Python has two primitive loop commands:

- While loops
- For loops

Python Arrays

- An array is a special variable, which can hold more than one value at a time.
An array is a special variable, which can hold more than one value at a time.

- Access the Elements of an Array
- The Length of an Array
- Looping Array Elements
- Adding Array Elements
- Removing Array Elements

Python Lambda

- A lambda function is a small anonymous function.
- A lambda function can take any number of arguments, but can only have one expression.

Python Classes and Objects

- Python is an object oriented programming language.
Almost everything in Python is an object, with its properties and methods.
A Class is like an object constructor, or a "blueprint" for creating objects.

Python Inheritance

- Inheritance allows us to define a class that inherits all the methods and properties from another class.
- **Parent class** is the class being inherited from, also called base class.
- **Child class** is the class that inherits from another class, also called derived class.

Python Iterators

- An iterator is an object that contains a countable number of values.

- An iterator is an object that can be iterated upon, meaning that you can traverse through all the values.
- Technically, in Python, an iterator is an object which implements the iterator protocol, which consist of the methods `iter ()` and `__next__()`.

Python Scope

- A variable created inside a function belongs to the local scope of that function, and can only be used inside that function.
- Local Scope
- Global Scope

Python Modules

- Consider a module to be the same as a code library.
- A file containing a set of functions you want to include in your application.

Python Datetime

- A date in Python is not a data type of its own, but we can import a module named `datetime` to work with dates as date objects.

Python Math

- Python has a set of built-in math functions, including an extensive `math` module, that allows you to perform mathematical tasks on numbers.

Python File Open

- File handling is an important part of any web application.
- Python has several functions for creating, reading, updating, and deleting files.

Python Delete File

- To delete a file, you must import the OS module, and run its `os.remove()` function:

Python File Write

- Write to an Existing File

To write to an existing file, you must add a parameter to the `open()` function:

"a" - Append - will append to the end of the file

"w" - Write - will overwrite any existing content

- Create a New File

To create a new file in Python, use the `open()` method, with one of the following parameters:

"x" - Create - will create a file, returns an error if the file exist

"a" - Append - will create a file if the specified file does not exist

"w" - Write - will create a file if the specified file does not exist

4.0 Introduction to MySQL.

MySQL is an open-source, fast reliable, and flexible relational database management system, typically used with PHP. This chapter is an introductory chapter about MySQL, what is MySQL, and the main features of MySQL are described here.

What is MySQL?

- MySQL is a database system used for developing web-based software applications.
- MySQL used for both small and large applications.
- MySQL is a relational database management system (RDBMS).
- MySQL is fast, reliable, and flexible and easy to use.
- MySQL supports standard SQL (Structured Query Language).
- MySQL is free to download and use.
- MySQL was developed by Michael Widenius and David Axmark in 1994.
- MySQL is presently developed, distributed, and supported by Oracle Corporation.
- MySQL Written in C, C++.

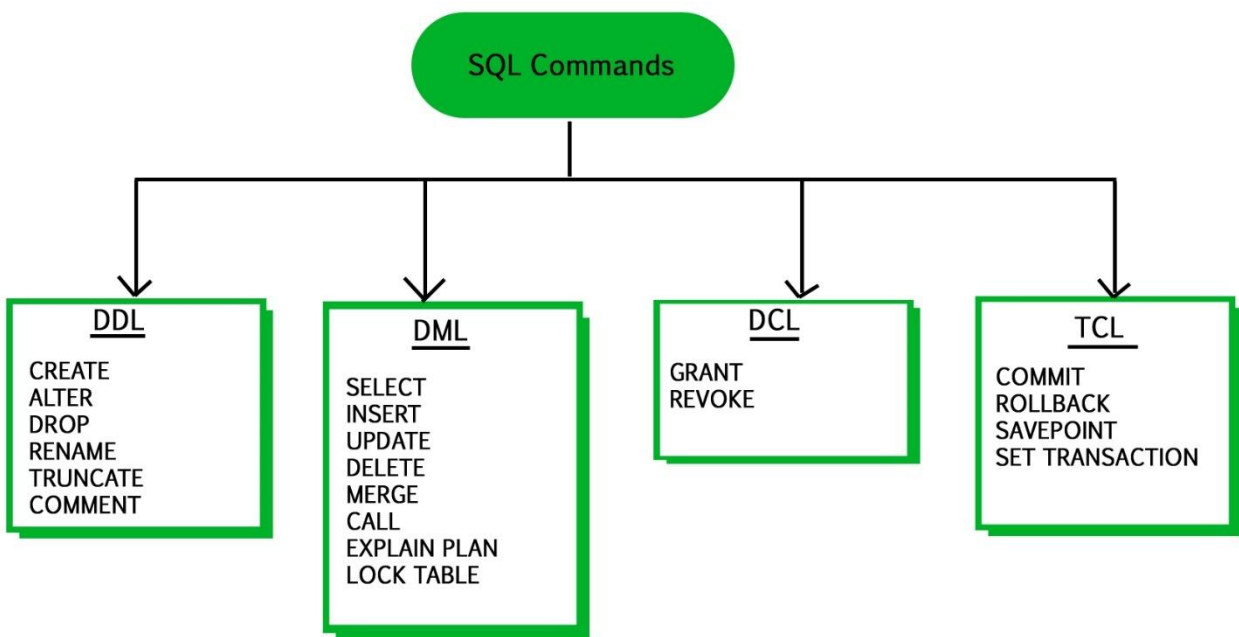
Main Features of MySQL

- MySQL server design is multi-layered with independent modules.
- MySQL is fully multithreaded by using kernel threads. It can handle multiple CPUs if they are available.
- MySQL provides transactional and non-transactional storage engines.

- MySQL has a high-speed thread-based memory allocation system.
- MySQL supports in-memory heap table.
- MySQL Handles large databases.
- MySQL Server works in client/server or embedded systems.
- MySQL Works on many different platforms.

MySQL commands

- SQL commands are divided into four subgroups, DDL, DML, DCL, and TCL.



MySQL CREATE Database and Tables.

- Create a database

CREATE DATABASE statement is used to create a database in MySQL.

We have to add the CREATE DATABASE statement to the mysqli_query() function to execute the command.

- Create a table in database

CREATE TABLE statement is used to create a table in MySQL database.

We have to add the CREATE TABLE statement to the mysqli_query() function to execute the command.

MySQL User Management

- MySQL Create User
- MySQL Drop User
- MySQL Show Users
- Change User Password

MySQL Database

- MySQL Create Database
- MySQL Select Database
- MySQL Show Databases
- MySQL Drop Database
- MySQL Copy Database

MySQL Table & Views

- MySQL CREATE Table
- MySQL ALTER Table
- MySQL Show Tables
- MySQL Rename Table
- MySQL TRUNCATE Table
- MySQL Describe Table
- MySQL DROP Table
- MySQL Temporary Table
- MySQL Copy Table
- MySQL Add/Delete Column
- MySQL Show Columns
- MySQL Table Locking
- MySQL Account Lock
- MySQL Account Unlock

MySQL Queries

- MySQL Queries
- MySQL Constraints
- MySQL INSERT Record
- MySQL UPDATE Record
- MySQL DELETE Record
- MySQL SELECT Record
- MySQL ReplaceInsert On Duplicate Key Update
- MySQL INSERT IGNORE

MySQL Indexes

- MySQL Create Index
- MySQL Drop Index
- MySQL Show Indexes
- MySQL Unique Index
- MySQL Clustered Index
- MySQL Clustered vs Non-Clustered Index

MySQL Clauses

- MySQL WHERE
- MySQL DISTINCT
- MySQL FROM
- MySQL ORDER BY
- MySQL GROUP BY
- MySQL HAVING

MySQL Conditions

- MySQL AND
- MySQL OR
- MySQL AND OR
- MySQL Boolean
- MySQL ANY
- MySQL Exists

- MySQL NOT
- MySQL Not Equal
- MySQL IS NULL
- MySQL IS NOT NULL
- MySQL BETWEEN

MySQL Join

- MySQL JOIN
- MySQL Inner Join
- MySQL Left Join
- MySQL Right Join
- MySQL CROSS JOIN
- MySQL SELF JOIN
- MySQL DELETE JOIN
- MySQL Update Join
- MySQL EquiJoin
- MySQL Natural JoinLeft Join vs Right Join
- MySQL Union vs Join

MySQL Key

- MySQL Unique Key
- MySQL Primary Key
- MySQL Composite Key

Aggregate Functions

- MySQL Aggregate Functions
- MySQL count()
- MySQL sum()
- MySQL avg()
- MySQL min()
- MySQL max()
- MySQL GROUP_CONCAT()
- MySQL first()
- MySQL last()

MySQL Misc

- MySQL Comments
- Export & Import Database
- Import CSV File in Database
- Export Table to CSV
- MySQL Subquery
- MySQL Derived Table
- MySQL UIDLEAD And LAG Function
- MySQL CTE
- MySQL On Delete Cascade
- MySQL UPSERT
- MySQL Commands Cheat-Sheet
- MySQL Transaction

- MySQL Partitioning
- MySQL ROW_NUMBER()
- MySQL Cursor
- MySQL Limit
- MySQL Stored FunctionNumber Format Function
- MySQL Ranking Functions
- MySQL Window Functions
- MySQL UnionUnion vs Union All
- MySQL VARCHAR
- MySQL ENUM
- MySQL SET
- MySQL DECIMAL
- MySQL Convert()
- MySQL COALESCE()
- MySQL Wildcards
- MySQL Alias
- MySQL Functions
- MySQL INT
- MySQL Today
- MySQL ROW COUNTPrepared Statement
- MySQL Literals(Constants)
- MySQL Date & Time
- MySQL Procedure
- MySQL Storage Engines
- MySQL Explain
- MySQL Sequence

- MySQL JSON
- MySQL Standard Deviation
- How to select nth Highest Record
- Find Duplicate Records
- Delete Duplicate Records
- MySQL Select Random Records
- MySQL Extract
- MySQL ProcessList
- MySQL BITChange Column Type
- MySQL Reset Auto-Increment

5.0 Description of project

In our industrial training, we got Project Titled “Database Queries”. Our group got our project summary which contains info like some questions, queries related to our database, questions related to a certain database, etc. This project is based on things that we have learnt in Industrial Training. Our project contains 18 Questions based on certain databases and some theory questions related to our topic.

The project addresses the following learning outcomes for the unit:

1. Demonstrate an understanding of the working of an database.
2. Demonstrate practical skills in data updation in a database.
3. Demonstrate and Develop entities (tables) with correct attributes included.

Our 1st three Questions were Theory Questions related to Key,Data types and Constraints.

1. What type of integrity is enforced when a primary key is declared?

➔ Referential integrity is usually enforced by the combination of a primary key or candidate key (alternate key) and a foreign key. For referential integrity to hold, any field in a table that is declared a foreign key can contain only values from a parent table's primary key or a candidate key.

2. Explain why it might be more appropriate to declare an attribute that contains only digits as a character data type instead of a numeric data type.

➔ It is important to note that an attribute that holds only the digits should be properly defined a character Data when the values are nominal because it is not treated as numbers (i.e) it is not possible to add or subtract nominal values. It does not have numerical significance but they serve as labels such as ZIP code, telephone numbers.

3. What is the difference between a column constraint and a table constraint?

➔ Column Level Constraints refers Only One Column .Which does not have the column name at declaration time.

Ex : Create table emp
(empno number primary Key,

```
--,  
--);
```

Table Level constraints refers one or more than one column . Which does have column name at declaration time.

Ex: Create table emp
(empno number ,

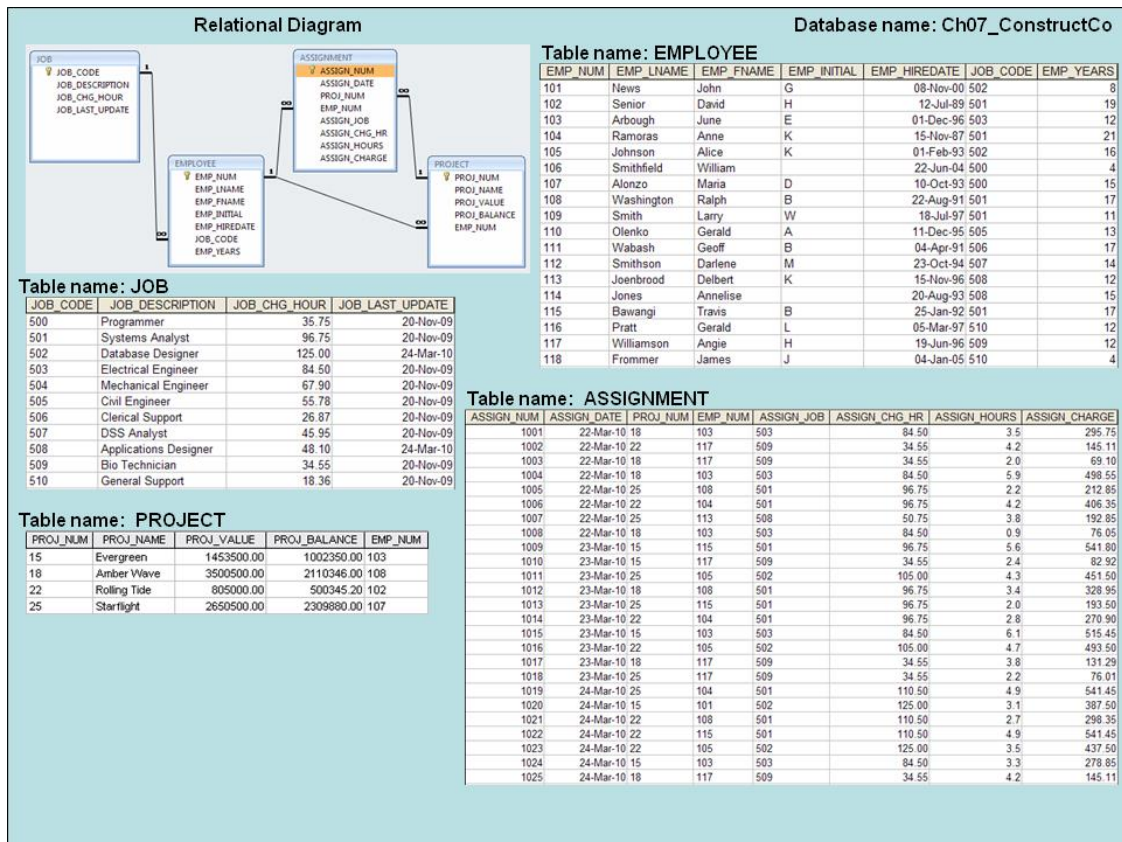
```
--,  
--,  
Constraints pk_con primary Key(empno));
```

All constraints can be created as Column and Table level except Not Null constraints.

Not Null constraints can only be created as column level constraints.
Does not have constraints name.

Then our Next set of question were based on a certain database named **Ch07_ConstructCo**.

The **Ch07_ConstructCo** database stores data for a consulting company that tracks all charges to projects. The charges are based on the hours each employee works on each project. The structure and contents of the **Ch07_ConstructCo** database are shown in the Figure below.



Note that the ASSIGNMENT table stores the JOB_CHG_HOUR values as an attribute (ASSIGN_CHG_HR) to maintain historical accuracy of the data. The JOB_CHG_HOUR values are likely to change over time. In fact, a JOB_CHG_HOUR change will be reflected in the ASSIGNMENT table. And, naturally, the employee primary job assignment might change, so the ASSIGN_JOB is also stored. Because those attributes are required to maintain the historical accuracy of the data, they are *not* redundant.

Given the structure and contents of the Ch07_ConstructCo database shown above, use SQL commands to answer Questions 4–18

- Write the SQL code that will create the table structure for a table named EMP_1. This table is a subset of the EMPLOYEE table. The basic EMP_1 table structure is summarized in the table below. (Note that the JOB_CODE is the FK to JOB.)

| ATTRIBUTE (FIELD) NAME | DATA DECLARATION |
|------------------------|------------------|
| EMP_NUM | CHAR(3) |
| EMP_LNAME | VARCHAR(15) |
| EMP_FNAME | VARCHAR(15) |
| EMP_INITIAL | CHAR(1) |
| EMP_HIREDATE | DATE |
| JOB_CODE | CHAR(3) |

➔ Program to create table emp_1

```
CREATE TABLE `proj`.`emp_1` (
  EMP_NUM` CHAR(3) NOT NULL,
  EMP_LNAME` VARCHAR(45) NOT NULL,
  EMP_FNAME` VARCHAR(45) NOT NULL,
  EMP_INITIAL` CHAR(1) NULL,
  EMP_HIREDATE` DATE NOT NULL,
  JOB_CODE` CHAR(3) NULL,
  PRIMARY KEY (`EMP_NUM`));
```

Creating emp_1 table which is the subset of the employee table

```
INSERT INTO emp_1 ( EMP_NUM, EMP_LNAME, EMP_FNAME,
  EMP_INITIAL,
  EMP_HIREDATE)
  SELECT EMP_NUM, EMP_LNAME, EMP_FNAME, EMP_INITIAL,
  EMP_HIREDATE
  FROM employee;
```

5. Having created the table structure in Question 4, write the SQL code to enter the first two rows for the table shown in Figure below.

| EMP_NUM | EMP_LNAME | EMP_FNAME | EMP_INITIAL | EMP_HIREDATE | JOB_CODE |
|---------|------------|-----------|-------------|--------------|----------|
| 101 | News | John | G | 08-Nov-00 | 502 |
| 102 | Senior | David | H | 12-Jul-89 | 501 |
| 103 | Arbough | June | E | 01-Dec-96 | 500 |
| 104 | Ramoras | Anne | K | 15-Nov-87 | 501 |
| 105 | Johnson | Alice | K | 01-Feb-93 | 502 |
| 106 | Smithfield | William | | 22-Jun-04 | 500 |
| 107 | Alonzo | Maria | D | 10-Oct-93 | 500 |
| 108 | Washington | Ralph | B | 22-Aug-91 | 501 |
| 109 | Smith | Larry | W | 18-Jul-97 | 501 |

➔ INSERT INTO

```
emp_1 (EMP_NUM, EMP_LNAME, EMP_FNAME, EMP_INITIAL,
EMP_HIREDATE, JOB_CODE)
VALUES ('101', 'News', 'John', 'G', '2000-11-08', '502'),
('102', 'Senior', 'David', 'H', '1989-07-12', '501');
```

6. Assuming the data shown in the EMP_1 table have been entered, write the SQL code that will list all attributes for a job code of 502.

➔ SELECT * FROM emp_1
WHERE JOB_CODE = 502;

7. Write the SQL code that will save the changes made to the EMP_1 table.

➔ COMMIT ;

8. Write the SQL code to change the job code to 501 for the person whose employee number (EMP_NUM) is 107. After you have completed the task, examine the results, and then reset the job code to its original value.

➔ UPDATE emp_1
SET JOB_CODE = '501'
WHERE EMP_NUM = '107';

To see the changes

```
SELECT * FROM emp_1
WHERE EMP_NUM = '107';
```

To Reset use

```
ROLLBACK;
```

9. Write the SQL code to delete the row for the person named William Smithfield, who was hired on June 22 2004, and whose job code classification is 500. (Hint: Use logical operators to include all of the information given in this problem.)

**➔ DELETE FROM emp_1
WHERE EMP_LNAME = 'Smithfield'
AND EMP_FNAME = 'William'
AND EMP_HIREDATE = '2004-June-22'
AND JOB_CODE = ' 500';**

10. Write the SQL code that will restore the data to its original status; that is, the table should contain the data that existed before you made the changes in Problems 5 and 6.

➔ ROLLBACK ;

11. Write the SQL code to create a copy of EMP_1, naming the copy EMP_2. Then write the SQL code that will add the attributes EMP_PCT and PROJ_NUM to its structure. The EMP_PCT is the bonus percentage to be paid to each employee. The new attribute characteristics are:

EMP_PCTNUMBER(4,2)

PROJ_NUMCHAR(3)

(Note: If your SQL implementation allows it, you may use DECIMAL(4,2) rather than NUMBER(4,2).)

**➔ CREATE TABLE emp_2 AS SELECT * FROM emp_1;
ALTER TABLE emp_2
ADD (EMP_PCT NUMBER(4,2)),
ADD (PROJ_NUM CHAR(3));**

12. Write the SQL code to change the EMP_PCT value to 3.85 for the person whose employee number (EMP_NUM) is 103. Next, write the SQL command sequences to change the EMP_PCT values as shown in the table below.

The contents of the EMP_2 table

| EMP_NUM | EMP_LNAME | EMP_FNAME | EMP_INITIAL | EMP_HIREDATE | JOB_CODE | EMP_PCT | PROJ_NUM |
|---------|------------|-----------|-------------|--------------|----------|---------|----------|
| 101 | News | John | G | 08-Nov-00 | 502 | 5.00 | |
| 102 | Senior | David | H | 12-Jul-89 | 501 | 8.00 | |
| 103 | Arbough | June | E | 01-Dec-96 | 500 | 3.85 | |
| 104 | Ramoras | Anne | K | 15-Nov-87 | 501 | 10.00 | |
| 105 | Johnson | Alice | K | 01-Feb-93 | 502 | 5.00 | |
| 106 | Smithfield | William | | 22-Jun-04 | 500 | 6.20 | |
| 107 | Alonzo | Maria | D | 10-Oct-93 | 500 | 5.15 | |
| 108 | Washington | Ralph | B | 22-Aug-91 | 501 | 10.00 | |
| 109 | Smith | Larry | W | 18-Jul-97 | 501 | 2.00 | |

➔ UPDATE EMP_2
SET EMP_PCT = 3.85
WHERE EMP_NUM = '103'

To Enter the remaining EMP_PCT values, use the following SQL statements

UPDATE EMP_2
SET EMP_PCT = 5.00
WHERE EMP_NUM = '101';

UPDATE EMP_2
SET EMP_PCT = 8.00
WHERE EMP_NUM = '102';

UPDATE EMP_2
SET EMP_PCT = 3.85
WHERE EMP_NUM = '103';

UPDATE EMP_2
SET EMP_PCT = 10.00
WHERE EMP_NUM = '104';

```
UPDATE EMP_2  
SET EMP_PCT = 5.00  
WHERE EMP_NUM = '105';
```

```
UPDATE EMP_2  
SET EMP_PCT = 6.20  
WHERE EMP_NUM = '106';
```

```
UPDATE EMP_2  
SET EMP_PCT = 5.15  
WHERE EMP_NUM = '107';
```

```
UPDATE EMP_2  
SET EMP_PCT = 10.00  
WHERE EMP_NUM = '108';
```

```
UPDATE EMP_2  
SET EMP_PCT = 2.00  
WHERE EMP_NUM = '109';
```

- 13. Using a single command sequence, write the SQL code that will change the project number (PROJ_NUM) to 18 for all employees whose job classification (JOB_CODE) is 500.**

➔

```
UPDATE EMP_2  
SET PROJ_NUM = '18'  
WHERE JOB_CODE = '500';
```

- 14. Using a single command sequence, write the SQL code that will change the project number (PROJ_NUM) to 25 for all employees whose job classification (JOB_CODE) is 502 or higher. When you finish Problems 10 and 11, the EMP_2 table will contain the data shown in the table below. (You may assume that the table has been saved again at this point.)**

The EMP_2 table contents after the modification

| EMP_NUM | EMP_LNAME | EMP_FNAME | EMP_INITIAL | EMP_HIREDATE | JOB_CODE | EMP_PCT | PROJ_NUM |
|---------|------------|-----------|-------------|--------------|----------|---------|----------|
| 101 | News | John | G | 08-Nov-00 | 502 | 5.00 | 25 |
| 102 | Senior | David | H | 12-Jul-89 | 501 | 8.00 | |
| 103 | Arbough | June | E | 01-Dec-96 | 500 | 3.85 | 18 |
| 104 | Ramoras | Anne | K | 15-Nov-87 | 501 | 10.00 | |
| 105 | Johnson | Alice | K | 01-Feb-93 | 502 | 5.00 | 25 |
| 106 | Smithfield | William | | 22-Jun-04 | 500 | 6.20 | 18 |
| 107 | Alonzo | Maria | D | 10-Oct-93 | 500 | 5.15 | 18 |
| 108 | Washington | Ralph | B | 22-Aug-91 | 501 | 10.00 | |
| 109 | Smith | Larry | W | 18-Jul-97 | 501 | 2.00 | |

➔ Update EMP_2

Set PROJ_NUM = '25'

WHERE JOB_CODE = '502' ;

15. Write the SQL code that will change the PROJ_NUM to 14 for those employees who were hired before January 1, 1994 and whose job code is at least 501. (You may assume that the table will be restored to its condition preceding this question.)

➔ UPDATE EMP_2

SET PROJ_NUM = '14'

WHERE EMP_HIREDATE <= '01-Jan-94'

AND JOB_CODE >= '501';

16. Write the two SQL command sequences required to:

a. Create a temporary table named TEMP_1 whose structure is composed of the EMP_2 attributes EMP_NUM and EMP_PCT.

b. Copy the matching EMP_2 values into the TEMP_1 table.

➔ A. CREATE TABLE temp_1

AS SELECT EMP_NUM, EMP_PCT, FROM emp_2;

B. CREATE TABLE temp_1 AS

EMP_NUM CHAR(3),

```
EMP_PCT    NUMBER(4,2));  
INSERT INTO temp_1  
SELECT EMP_NUM, EMP_PCT FROM emp_2;
```

17. Write the SQL command that will delete the newly created TEMP_1 table from the database.

➔ DROP TABLE TEMP_1;

18. Write the SQL code required to list all employees whose last names start with *Smith*. In other words, the rows for both Smith and Smithfield should be included in the listing. Assume case sensitivity.

➔ SELECT *
FROM EMP_2
WHERE EMP_LNAME LIKE 'Smith%';