**Institute of Information Technology & Management**

**DepartmentofComputerScience**

LabManual

**DATABASEMANAGEMENTSYSTEM**

**MCA-I (Sem)**

**SubjectCode-**

**Subject Teacher:**

Dr. Pankaj Kumar Varshney

Associate Professor

**DOs and DON’Ts in Laboratory:**

1. Make entry in the Log Book as soon as you enter the Laboratory.

2. All the students should sit according to their roll numbers starting from their left to right.

3. All the students are supposed to enter the terminal number in the log book.

4. Do not change the terminal on which you are working.

5. All the students are expected to get at least the algorithm of the program/concept to be implemented.

6. Strictly observe the instructions given by the teacher/Lab Instructor.

7. Do not disturb machine Hardware / Software Setup.

**Instruction for Laboratory Teachers:**

1. Submission related to whatever lab work has been completed should be done during the next lab session along with signing the index.

2. The promptness of submission should be encouraged by way of marking and evaluation patterns that will benefit the sincere students.

3. Continuous assessment in the prescribed format must be followed.

**Lesson Plan for Data Base Management System Lab**

**PROGRAMME: MCA SEMESTER: I PAPER CODE: ACADEMIC SESSION: 2023-24**

**Course Objective:**

* To understand the basics concepts of Oracle.
* To understand and creation of Database and tables in RDBMS
* To implement different RDBMS Constraints and queries to design database and manipulate

data in it

* To implements physical design of a database.

**Course Outcomes:**

**CO1.** To give basic Introduction to Oracle

**CO2.**To implement different DDL commands to create database.

**CO3.**To implement different DDL commands along with Table constraint.

**CO4.**To implements DML Commands and query to create database.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Course Outcomes** | **Programme Outcomes** | | | | |
|  | **PO1**  **Advanced Subject knowledge** | **PO2**  **Enquiry-based learning** | **PO3**  **Cognitive skills and critical thinking** | **PO4**  **Communication, Adaptive & Interactional Skills** | **PO5**  **Holistic Outlook** |
| **CO1** | **LOW** | **MODERATE** | **MODERATE** | **MODERATE** | **HIGH** |
| **CO2** | **MODERATE** | **MODERATE** | **MODERATE** | **LOW** | **MODERATE** |
| **CO3** | **MODERATE** | **HIGH** | **HIGH** | **HIGH** | **HIGH** |
| **CO4** | **HIGH** | **MODERATE** | **MODERATE** | **MODERATE** | **HIGH** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.NO | TITLE |  |  |  |
|  | The following are two suggestive databases. The students may use any one or both databases for their core practical.   1. **COLLEGE DATABASE:**   STUDENT (USN, SName, Address, Phone, Gender)  SEMSEC (SSID, Sem, Sec)  CLASS(USN, SSID)  SUBJECT (Subcode, Title, Sem, Credits\_  IAMARKS (USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)   1. **COMPANY DATABASE:**   EMLPOYEE (SSN, Name, Address, Sex, Salary, SuperSSN, Dno.)  DEPARTMENT (DNo, Dname, MgrSSN, MgrStartDate)  DLOCATION (DNO, DLOC)  PROJECT (PNo, PName, PLocation, DNo)  WORKS ON (SSN, PNo, Hours) |  |  |  |
| 1. | Draw an E-R diagram from given entities and their attributes |  |  |  |
| 2. | Convert the E-R diagram into a Relational model with proper constraints. |  |  |  |
| 3. | Write queries to execute following DDL commands :  CREATE: Create the structure of a table with at least five columns  ALTER: Change the size of a particular column.  Add a new column from the existing table.  Remove a column from the table.  DROP: Destroy the table along with its data. |  |  |  |
| 4. | Write queries to execute following DML Commands:  INSERT: Insert five records in each table.  UPDATE: modify data in single and multiple columns in a table  DELETE: Delete selective and all record from the table. |  |  |  |
| 5. | Write queries to execute following DML Commands:  SELECT: Retrieve the entire contents of the table.  Retrieve the selective contents (based on provided conditions) from a table.  Retrieve contents from a table based on various operators i.e. string operators, logical operators and conditional operators, Boolean operators.  Sort the data in ascending and descending order in a table on the basis of one column or more than one column. |  |  |  |
| 6. | Create table using following integrity constraints.  Primary Key  Unique key  Not Null  Check  Default  Foreign Key |  |  |  |
| 7. | Write queries to execute following Aggregate functions  Sum, Avg, Count, Minimum and Maximum value of a numeric column of a table using aggregate function |  |  |  |
| 8. | Retrieve data from a table using alias names. |  |  |  |
| 9. | Retrieve data of a table using nested queries. |  |  |  |
| 10. | Retrieve data from more than one table using inner join, left outer, right outer and full outer joins. |  |  |  |
| 11. | Create view from one table and more one table. |  |  |  |
| 12. | Create index on a column of a table. |  |  |  |
| 13. | Consider the Insurance company’s Database given below. The primary keys are underlined and the data typesare specified.  PERSON(driver\_id# : string, name : string, address : string)  CAR(regno : string, model : string, year : int)  ACCIDENT (report\_number : int, acc\_date : date, location : string)  OWNS(driver\_id# :string, regno : string)  PARTICIPATED(driver\_id# :string, regno : string, report\_number : int, damage\_amount : number(10,2))   1. Create the above table by properly specified the primary key and the foreign key 2. Enter at least five tuples for each relation. 3. Demonstrate how you can 4. Update the damage amount for the car with a specificregno, the accident with report number 12 to 25000. 5. Add a new accident to the database. 6. Find the total number of people who owned cars that were involved in accident in 2002 7. Find the number of accident in which cars belonging to a specific model were involved. |  |  |  |
| 14. | Consider the following schema of a library management system, Write the SQL queries for the questions given below;  **Student(Stud\_no : inter, Stud\_name: string)**  **Membership(Mem\_no : integer, Stud\_no: integer)**  **Book\_(book\_no: integer, book\_name: string, author: string)**  **Iss\_rec\_(iss\_no:integer, iss\_date:date, Mem\_no:integer, book\_no:integer)**   1. Create the table with the appropriate integrity constraints. 2. Insert around 10 records in each of the tables. 3. Display all records for the tables. 4. List all the student names with their membership numbers. 5. List all the issues for the current data with student and Book Names. 6. List the details of students who borrowed book whose author is Elmarsi&Navathe 7. Give a count of how many books have been bought by each student. 8. Give a list of book taken by student with stud\_no as 1005 9. Delete the list of book details which are issued as of today. 10. Create a view which list out the iss\_no, iss\_date, stud\_name, book name |  |  |  |
| 15. | Use the relations below to write SQL queries to solve the business problems specified.  CLINET (clientno#,name, client\_referred\_by#)  ORDER (orderno#, clientno#, order\_date, empid#)  ORDER\_LINE (orderno#, order line number#, item\_number#, no\_of\_items, item\_cost,shipping\_date)  ITEM (item\_number#, item\_type,cost)  EMPLOYEE (empid#, emp\_type#, deptno, salary, firstname, lastname)  Note:   1. Column followed by # is the primary key of the table. 2. Each client may be referred by another client. If so the client number of the referring client is stored in referred\_by. 3. The total cost for a particular order line=no\_of\_items \* item\_cost.c.   Write queries for the following   1. Create all the above tables. 2. Insert at least five records. 3. Display all the rows and column in the CLIENT table. Sort by client name in reverse alphabetical order. 4. Display all the item number and total cost for each order line (total cost= no of items X items cost). Name the calculated column TOTAL COST. 5. Display all the client number in the ORDER table. Remove duplicates. 6. Display the order number and client number from the ORDER table. Output the result in the format. Client <clientno>ordered<orderno> 7. Display full details from the ORDER\_LINE table where the item number is (first condition) between 1 and 200 (no> or < operators) OR the item number is greater than 1000 AND (second condition) the item cost is not in the list 1000, 2000, 3000 OR the order number is not equal to 1000. 8. Display the client name and order date for all orders. 9. Repeat query (6) but also display all clients who have never ordered anything. 10. Display the client name and order date for all orders using the join keywords. 11. Display the client name and order date for all orders using the JOIN method. 12. Display the client number, order date and shipping date for all orders where the shipping date is between three and six months after the order date. 13. Display the client number and name and the client number and name of the person who referred that client. 14. Display the client name in upper case only and in lower case only. 15. Display the second to fifty characters in each client name. |  |  |  |

# Theory:

# Oracle has many tools such as SQL \* PLUS, Oracle Forms, Oracle Report Writer, Oracle Graphics etc.

# SQL \* PLUS: The SQL \* PLUS tool is made up of two distinct parts. These are

# Interactive SQL: Interactive SQL is designed for create, access and manipulate data structures like tables and indexes.

# PL/SQL: PL/SQL can be used to developed programs for different applications.

# Oracle Forms: This tool allows you to create a data entry screen along with the suitable menu objects. Thus it is the oracle forms tool that handles data gathering and data validation in a commercial application.

# Report Writer: Report writer allows programmers to prepare innovative reports using data from the oracle structures like tables, views etc. It is the report writer tool that handles the reporting section of commercial application.

# Oracle Graphics: Some of the data can be better represented in the form of pictures.

# The oracle graphics tool allows programmers to prepare graphs using data from oracle structures like tables, views etc.

# SQL (Structured Query Language):

# Structured Query Language is a database computer language designed for managing data in relational database management systems (RDBMS), and originally based upon Relational Algebra. Its scope includes data query and update, schema creation and modification, and data access control.

# SQL was one of the first languages for Edgar F. Codd's relational model and became the most widely used language for relational databases.

# IBM developed SQL in mid of 1970’s.

# Oracle incorporated in the year 1979.

# SQL used by IBM/DB2 and DS Database Systems.

# SQL adopted as standard language for RDBS by ASNI in 1989.

# DATA TYPES:

# CHAR (Size): This data type is used to store character strings values of fixed length.

# The size in brackets determines the number of characters the cell can hold. The maximum number of character is 255 characters.

# VARCHAR (Size) / VARCHAR2 (Size): This data type is used to store variable length alphanumeric data. The maximum character can hold is 2000 character.

# NUMBER (P, S): The NUMBER data type is used to store number (fixed or floating point). Number of virtually any magnitude may be stored up to 38 digits of precision.

# Number as large as 9.99 \* 10 124. The precision (p) determines the number of places to the right of the decimal. If scale is omitted then the default is zero. If precision is omitted, values are stored with their original precision up to the maximum of 38 digits.

# DATE: This data type is used to represent date and time. The standard format is DD-MM-YY as in 17-SEP-2009. To enter dates other than the standard format, use the appropriate functions. Date time stores date in the 24-Hours format. By default the time in a date field is 12:00:00 am, if no time portion is specified. The default date for a date

# field is the first day the current month.

# LONG: This data type is used to store variable length character strings containing up to 2GB. Long data can be used to store arrays of binary data in ASCII format. LONG values cannot be indexed, and the normal character functions such as SUBSTR cannot be applied.

# RAW: The RAW data type is used to store binary data, such as digitized picture or image. Data loaded into columns of these data types are stored without any further conversion. RAW data type can have a maximum length of 255 bytes. LONG RAW data type can contain up to 2GB.

# SQL language is sub-divided into several language elements, including:

# Clauses, which are in some cases optional, constituent components of statements and queries.

# Expressions, which can produce either scalar values or tables consisting of columns and rows of data.

# Predicates which specify conditions that can be evaluated to SQL three-valued logic (3VL) Boolean truth values and which are used to limit the effects of statements and queries, or to change program flow.

# Queries which retrieve data based on specific criteria.

# Statements which may have a persistent effect on schemas and data, or which may control transactions, program flow, connections, sessions, or diagnostics.

# SQL statements also include the semicolon (";") statement terminator. Though not required on every platform, it is defined as a standard part of the SQL grammar.

* Insignificant white space is generally ignored in SQL statements and queries, making it easier to format SQL code for readability.

# Experiment1:

**ConsiderfollowingdatabasesanddrawERdiagramandconvertentitiesandrelationshipstorelationtableforagivenscenario.**

# COLLEGEDATABASE:

STUDENT(*USN,SName,Address, Phone, Gender*)SEMSEC(*SSID,Sem,Sec*)

CLASS(*USN,SSID*)

SUBJECT(*Subcode,Title,Sem,Credits*)

IAMARKS(*USN,Subcode,SSID,Test1,Test2,Test3,FinalIA*)

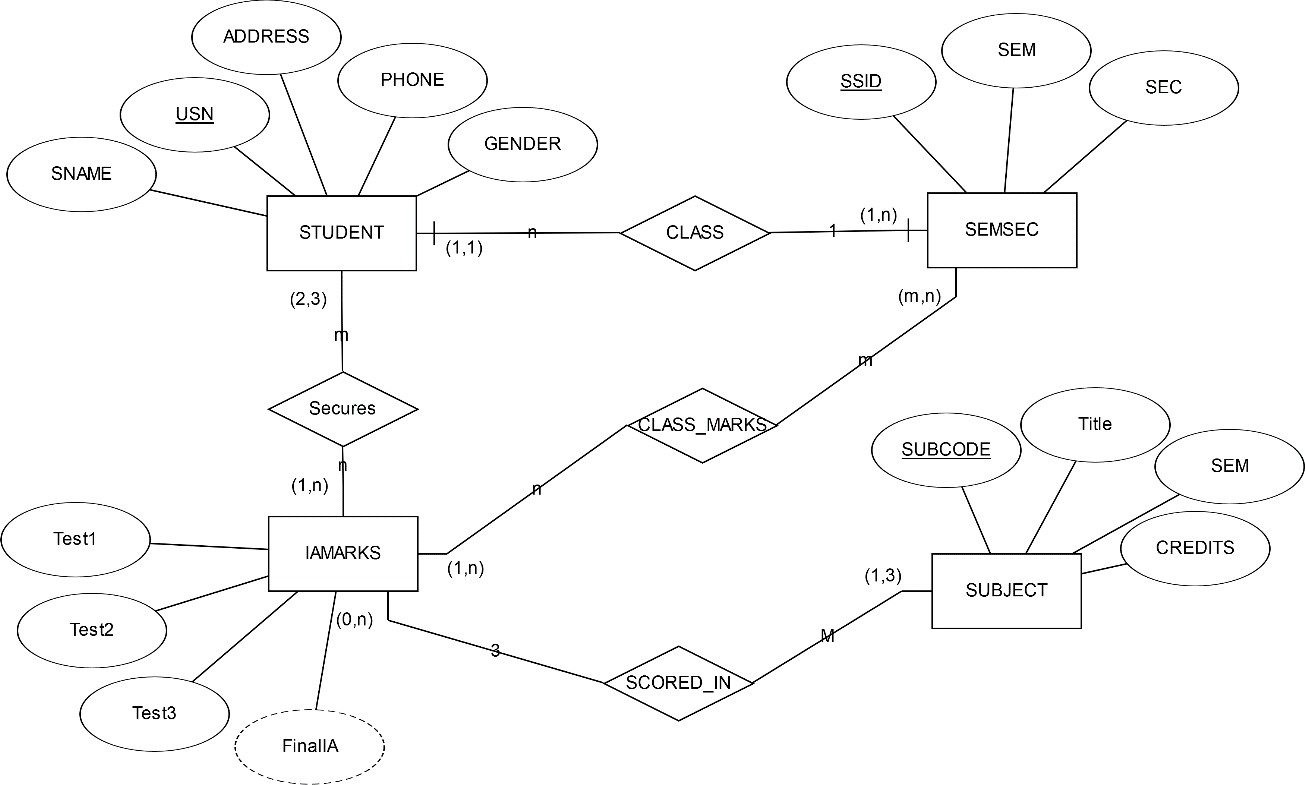
# COMPANYDATABASE:

EMPLOYEE (*SSN, Name, Address, Sex, Salary, SuperSSN, DNo*)DEPARTMENT(*DNo,DName,MgrSSN,MgrStartDate*)DLOCATION (*DNo,DLoc*)

PROJECT (*PNo, PName, PLocation, DNo*)WORKS\_ON (*SSN,PNo,Hours*)

# SOLUTION:

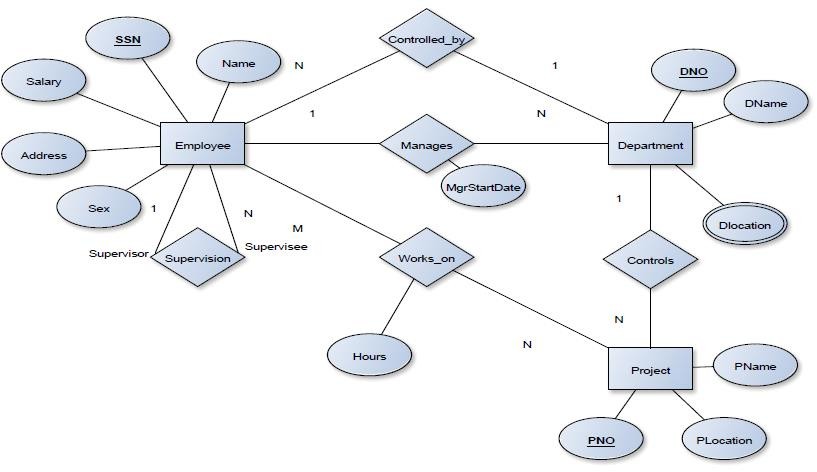
**CollegeDatabase: E-RDiagram**



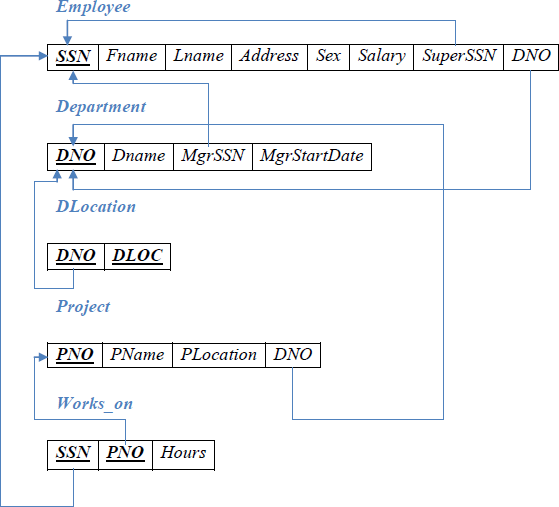
# Mappingentities andrelationshipsto relationtable(SchemaDiagram)

**COMPANYDATABASE:**

# E-RDiagram

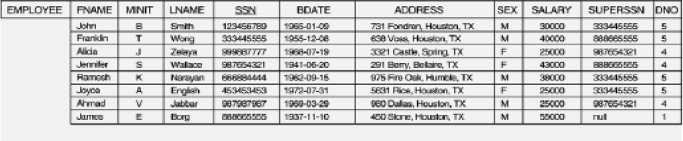


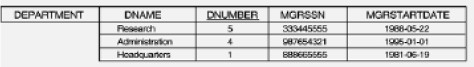
**SchemaDiagram**



# Experiment3

**ConsidertheCompany databasewithfollowingtables**





Performthefollowing:

* 1. Createcompanydatabase
  2. Viewingalldatabases
  3. ViewingallTablesinaDatabase,
  4. CreatingTables(WithandWithoutConstraints)
  5. Inserting/Updating/Deleting RecordsinaTable
  6. Saving(Commit)andUndoing(rollback)

# SOLUTION:

1. CreatingaDatabase

CREATEDATABASECompany;

1. Viewing all databasesSHOWDATABASES;
2. ViewingallTablesinaDatabase,SHOWtables;
3. CreatingTables(WithandWithoutConstraints)CREATETABLEDEPARTMENT

(DNOVARCHAR2(20)PRIMARYKEY,DNAMEVARCHAR2(20),MGRSTARTDATEDATE);

CREATETABLEEMPLOYEE

(SSNVARCHAR2(20)PRIMARYKEY,FNAMEVARCHAR2(20),

LNAMEVARCHAR2(20),

ADDRESSVARCHAR2(20),

SEX CHAR (1),SALARYINTEGER,

SUPERSSNREFERENCESEMPLOYEE(SSN),DNOREFERENCESDEPARTMENT(DNO));

**NOTE:**OnceDEPARTMENTandEMPLOYEEtablesarecreatedwemustalterdepartmenttable toaddforeignconstraintMGRSSNusingsqlcommand

ALTERTABLEDEPARTMENT

ADDMGRSSNREFERENCESEMPLOYEE(SSN);

1. Inserting/Updating/DeletingRecordsinaTable,

INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES(‗RNSECE01‘,‘JOHN‘,‘SCOTT‘,‘BANGALORE‘,‘M‘,450000);INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES(‗RNSCSE01‘,‘JAMES‘,‘SMITH‘,‘BANGALORE‘,‘M‘,500000);INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES(‗RNSCSE02‘,‘HEARN‘,‘BAKER‘,‘BANGALORE‘,‘M‘,700000);INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES(‗RNSCSE03‘,‘EDWARD‘,‘SCOTT‘,‘MYSORE‘,‘M‘,500000);

INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES (‗RNSCSE04‘,‘PAVAN‘,‘HEGDE‘,‘MANGALORE‘,‘M‘, 650000);INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES(‗RNSCSE05‘,‘GIRISH‘,‘MALYA‘,‘MYSORE‘,‘M‘,450000);

INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES (‗RNSCSE06‘,‘NEHA‘,‘SN‘,‘BANGALORE‘,‘F‘,800000);

INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES (‗RNSACC01‘,‘AHANA‘,‘K‘,‘MANGALORE‘,‘F‘,350000);

INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES (‗RNSACC02‘,‘SANTHOSH‘,‘KUMAR‘,‘MANGALORE‘,‘M‘, 300000);INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES (‗RNSISE01‘,‘VEENA‘,‘M‘,‘MYSORE‘,‘M‘,600000);

INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES (‗RNSIT01‘,‘NAGESH‘,‘HR‘,‘BANGALORE‘,‘M‘,500000);

INSERTINTODEPARTMENTVALUES(‗1‘,‘ACCOUNTS‘,‘01-JAN-01‘,‘RNSACC02‘);

INSERTINTODEPARTMENTVALUES(‗2‘,‘IT‘,‘01-AUG-16‘,‘RNSIT01‘);

INSERT INTO DEPARTMENT VALUES (‗3‘,‘ECE‘,‘01-JUN-08‘,‘RNSECE01‘);INSERT INTO DEPARTMENT VALUES (‗4‘,‘ISE‘,‘01-AUG-15‘,‘RNSISE01‘);INSERTINTODEPARTMENTVALUES(‗5‘,‘CSE‘,‘01-JUN-02‘,‘RNSCSE05‘);

Update

UPDATEEMPLOYEESETDNO=‘5‘,SUPERSSN=‘RNSCSE06‘WHERESSN=‘RNSCSE05‘;

Delete entriesofemployee table whereDNO=1;DELETEFROMEMPLOYEEWHEREDNO=1;

1. COMMITandROLLBACK

Before concluding this section on Data Manipulation Language commands there are twofurther commands, which are very useful. Changes made to the database by INSERT,UPDATE and DELETE commands are temporary until explicitly committed. This isperformedbythecommand:

# COMMIT;

Onexecutionofthiscommand allchangestothedatabasemadeby you aremadepermanentandcannotbeundone.

* + ACOMMITisautomaticallyexecutedwhenyouexitnormallyfromSQL\*Plus.However,itdoesnoharmto occasionallyissuea COMMITcommand.
  + ACOMMITdoesnotapplyto anySELECTcommandsasthereisnothingtocommit.
  + A COMMIT does not apply to any DDL commands (eg CREATE TABLE,CREATEINDEX,etc).Theseareautomaticallycommittedandcannotberolledback.
  + Ifyou wished to rollback(ieundo)anychangesmadetothedatabasesincethelastcommit,youcanissue the command:

# ROLLBACK;

Agroupofrelated SQLcommandsthatallhaveto completesuccessfullyorotherwiseberolledback,iscalled atransaction.PartofyourresearchforOutcome3includesinvestigatingtransactionprocessingand theimplicationsofrollback and commit.

# Experiment4

ConsiderDepttable

|  |  |  |
| --- | --- | --- |
| DEPTNO | DNAME | LOC |

Performthefollowing:

1. Renamethetabledeptasdepartment
2. Addanewcolumn PINCODEwithnotnullconstraintstotheexistingtableDEPT
3. Allconstraintsandviewsthatreferencethecolumnaredroppedautomatically, alongwiththecolumn.
4. RenamethecolumnDNAME toDEPT\_NAMEindepttable
5. ChangethedatatypeofcolumnlocasCHARwith size10
6. Deletetable

# SOLUTION:

**CreateTable**

SQL>CREATETABLEDEPT(DEPTNOINTEGER,DNAMEVARCHAR(10),LOCVARCHAR(4),PRIMARYKEY(DEPTNO));

1. Renamethetabledeptasdepartment

SQL>ALTERTABLEDEPT RENAMETODEPARTMENT;

Tablealtered.

1. Add a new column PINCODE with not null constraints to the existing table DEPTSQL>ALTERTABLEDEPARTMENTADD(PINCODENUMBER(6)NOTNULL);Table altered.

SQL>DESCDEPARTMENT;

Name Null? Type

DEPTNO NOTNULLNUMBER(38)DNAME VARCHAR2(10)

LOC VARCHAR2(4)

PINCODE NOTNULLNUMBER(6)

1. Allconstraintsandviewsthatreferencethecolumnaredroppedautomatically, alongwiththecolumn.

SQL>ALTERTABLEDEPARTMENTDROPcolumnLOCCASCADECONSTRAINTS;

Tablealtered.

SQL>descdept

Name Null? Type

DEPTNO NOTNULLNUMBER(38)DNAME VARCHAR2(10)

PINCODE NOTNULLNUMBER(6)

1. RenamethecolumnDNAMEtoDEPT\_NAMEindepttable

SQL>ALTERTABLEDEPTRENAMECOLUMNDNAME TODEPT\_NAME;

Tablealtered.

SQL>DESCDEPARTMENT;

Name Null? Type

DEPTNO NOTNULLNUMBER(38)DEPT\_NAME VARCHAR2(10)LOC VARCHAR2(4)

PINCODE NOTNULLNUMBER(6)

1. Changethe datatypeofcolunmloc asCHARwith size 10

SQL>ALTERTABLEDEPARTMENTMODIFYLOCCHAR(10);

Tablealtered.

SQL>DESCDEPARTMENT;

Name Null? Type

DEPTNO NOTNULLNUMBER(38)DEPT\_NAME VARCHAR2(10)LOC CHAR(10)

PINCODE NOTNULLNUMBER(6)

1. Deletetable

SQL>DROPTABLEDEPARTMENT;

Table dropped.

# Experiment5A

ConsiderEmployeetable

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EMPNO** | **EMP\_NAME** | **DEPT** | **SALARY** | **DOJ** | **BRANCH** |
| E101 | Amit | oduction | 45000 | 12-Mar-00 | Bangalore |
| E102 | Amit | HR | 70000 | 03-Jul-02 | Bangalore |
| E103 | sunita | anagemen | 120000 | 11-Jan-01 | mysore |
| E105 | sunita | IT | 67000 | 01-Aug-01 | mysore |
| E106 | mahesh | Civil | 145000 | 20-Sep-03 | Mumbai |

**Performthefollowing**

1. **Displayallthefieldsof employeetable**
2. **Retrieveemployeenumberandtheirsalary**
3. **Retrieveaveragesalaryof allemployee**
4. **Retrievenumberof employee**
5. **Retrievedistinctnumberofemployee**
6. **Retrievetotalsalaryofemployeegroupbyemployeenameandcountsimilarnames**
7. **Retrievetotalsalaryofemployeewhichisgreaterthan>120000**
8. **Displaynameofemployeeindescendingorder**
9. **DisplaydetailsofemployeewhosenameisAMITandsalarygreaterthan50000;**
10. **Displayallthefieldsof employeetable**

SQL>select\*fromemployee;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EMPNO** | **EMP\_NAME** | **DEPT** | **SALARY** | **DOJ** | **BRANCH** |
| E101 | Amit | Production | 45000 | 12-MAR-00 | Bangalore |
| E102 | Amit | HR | 70000 | 03-JUL-02 | Bangalore |
| E103 | sunita | Management | 120000 | 11-JAN-01 | mysore |
| E105 | sunita | IT | 67000 | 01-AUG-01 | mysore |
| E106 | mahesh | Civil | 145000 | 20-SEP-03 | Mumbai |

1. **Retrieveemployeenumberandtheirsalary**

SQL> select empno, salary from employee;EMPNO SALARY

|  |  |
| --- | --- |
| E101 | 45000 |
| E102 | 70000 |
| E103 | 120000 |
| E105 | 67000 |
| E106 | 145000 |

1. **Retrieveaveragesalaryof allemployee**

SQL>selectavg(salary)fromemployee;

AVG(SALARY)

89400

1. **Retrievenumberof employee**

SQL> select count(\*) from employee;COUNT(\*)

5

1. **Retrievedistinctnumberofemployee**

SQL>selectcount(DISTINCTemp\_name)fromemployee;COUNT(DISTINCTEMP\_NAME)

3

1. **Retrievetotalsalaryofemployee groupbyemployeenameandcountsimilarnames**

SQL>SELECTEMP\_NAME,SUM(SALARY),COUNT(\*)FROMEMPLOYEE2GROUPBY(EMP\_NAME);

|  |  |  |
| --- | --- | --- |
| EMP\_NAME | SUM(SALARY) | COUNT(\*) |
| mahesh | 145000 | 1 |
| sunita | 187000 | 2 |
| Amit | 115000 | 2 |

1. **Retrievetotalsalaryofemployeewhichisgreaterthan>120000**

SQL>SELECTEMP\_NAME,SUM(SALARY)FROMEMPLOYEE2GROUPBY(EMP\_NAME)

3HAVINGSUM(SALARY)>120000;

EMP\_NAME SUM(SALARY)

mahesh 145000

sunita 187000

1. **Displaynameof employeeindescendingorder**

SQL> select emp\_name from employee2orderbyemp\_namedesc;

EMP\_NAME

sunitasunitamaheshAmitAmit

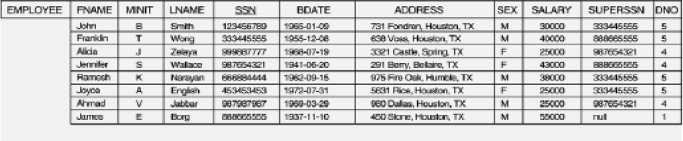
1. **DisplaydetailsofemployeewhosenameisAMITandsalarygreaterthan50000;**

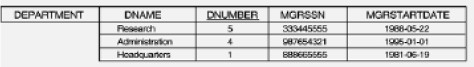
SQL>select\*fromemployee

2whereemp\_name='Amit'andsalary>50000;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| EMPNO | EMP\_NAME | DEPT | SALARY | DOJ | BRANCH |
| E102 | Amit | HR | 70000 | 03-JUL-02 | Bangalore |

# Experiment5B

**Foragiventables**



# Createtablesandperformthefollowing

1. Howtheresultingsalariesifeveryemployeeworkingonthe‘Research’Departmentsisgivena10percentraise.
2. Find the sumofthe salariesofallemployeesofthe‘Accounts’department,aswellasthemaximumsalary,theminimumsalary,and theaveragesalaryin thisdepartment
3. RetrievethenameofeachemployeeControlledbydepartmentnumber5(useEXISTSoperator).
4. Retrievethenameofeachdeptand number ofemployeesworkingineachdepartmentwhichhas atleast2employees
5. Retrievethenameofemployeeswhobornintheyear 1990’s
6. Retrievethenameofemployeesandtheirdeptname(usingJOIN)

# SOLUTION

SQL>CREATETABLEDEPARTMENT(

DNOVARCHAR2(20)PRIMARYKEY,DNAMEVARCHAR2(20),MGRSTARTDATEDATE);

SQL>DESCDEPARTMENT;

Name Null? Type

DNO NOTNULLVARCHAR2(20)DNAME VARCHAR2(20)

MGRSTARTDATE DATE

SQL>CREATETABLEEMPLOYEE(

SSNVARCHAR2(20)PRIMARYKEY,FNAMEVARCHAR2(20),

LNAMEVARCHAR2(20),

ADDRESSVARCHAR2(20),

SEX CHAR (1),SALARYINTEGER,

SUPERSSNREFERENCESEMPLOYEE(SSN),DNOREFERENCESDEPARTMENT(DNO));

SQL>DESCEMPLOYEE;

Name Null? Type

|  |  |  |
| --- | --- | --- |
| SSN | NOTNULL | VARCHAR2(20) |
| FNAME |  | VARCHAR2(20) |
| LNAME |  | VARCHAR2(20) |
| ADDRESS |  | VARCHAR2(20) |
| SEX |  | CHAR(1) |
| SALARY |  | NUMBER(38) |
| SUPERSSN |  | VARCHAR2(20) |
| DNO |  | NUMBER(38) |

SQL>ALTERTABLEDEPARTMENT

2ADDMGRSSNREFERENCESEMPLOYEE(SSN);Table altered.

SQL>DESCDEPARTMENT;

Name Null? Type

DNO NOT NULL VARCHAR2(20)DNAME VARCHAR2(20)

MGRSTARTDATE DATE

MGRSSN VARCHAR2(20)

INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES(‘RNSECE01‘,‘JOHN‘,‘SCOTT‘,‘BANGALORE‘,‘M‘,450000);

INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES(‘RNSCSE01‘,‘JAMES‘,‘SMITH‘,‘BANGALORE‘,‘M‘,500000);INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES (‘RNSCSE02‘,‘HEARN‘,‘BAKER‘,‘BANGALORE‘,‘M‘, 700000);INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES(‘RNSCSE03‘,‘EDWARD‘,‘SCOTT‘,‘MYSORE‘,‘M‘,500000);

INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES (‘RNSCSE04‘,‘PAVAN‘,‘HEGDE‘,‘MANGALORE‘,‘M‘, 650000);INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES (‘RNSCSE05‘,‘GIRISH‘,‘MALYA‘,‘MYSORE‘,‘M‘,450000);

INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES (‘RNSCSE06‘,‘NEHA‘,‘SN‘,‘BANGALORE‘,‘F‘,800000);

INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES (‘RNSACC01‘,‘AHANA‘,‘K‘,‘MANGALORE‘,‘F‘,350000);

INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES (‘RNSACC02‘,‘SANTHOSH‘,‘KUMAR‘,‘MANGALORE‘,‘M‘, 300000);INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES (‘RNSISE01‘,‘VEENA‘,‘M‘,‘MYSORE‘,‘M‘,600000);

INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES (‘RNSIT01‘,‘NAGESH‘,‘HR‘,‘BANGALORE‘,‘M‘,500000);

INSERTINTODEPARTMENTVALUES(1,‘ACCOUNTS‘,‘01-JAN-01‘,‘RNSACC02‘);

INSERT INTO DEPARTMENT VALUES (2,‘IT‘,‘01-AUG-16‘,‘RNSIT01‘);INSERTINTODEPARTMENTVALUES(3,‘ECE‘,‘01-JUN-08‘,‘RNSECE01‘);INSERT INTO DEPARTMENT VALUES (4,‘ISE‘,‘01-AUG-15‘,‘RNSISE01‘);INSERTINTODEPARTMENTVALUES(5,‘CSE‘,‘01-JUN-02‘,‘RNSCSE05‘);

**Note:updateentriesofemployeetabletofillmissingfieldsSUPERSSNandDNO**UPDATE EMPLOYEE SET SUPERSSN=NULL, DNO=‘3‘ WHERESSN=‘RNSECE01‘;

UPDATEEMPLOYEESETSUPERSSN=‘RNSCSE02‘,DNO=‘5‘WHERESSN=‘RNSCSE01‘;

UPDATEEMPLOYEESETSUPERSSN=‘RNSCSE03‘,DNO=‘5‘WHERESSN=‘RNSCSE02‘;

UPDATEEMPLOYEESETSUPERSSN=‘RNSCSE04‘,DNO=‘5‘WHERESSN=‘RNSCSE03‘;

UPDATE EMPLOYEE SET DNO=‘5‘, SUPERSSN=‘RNSCSE05‘ WHERESSN=‘RNSCSE04‘; UPDATE EMPLOYEE SET DNO=‘5‘, SUPERSSN=‘RNSCSE06‘WHERESSN=‘RNSCSE05‘;

UPDATEEMPLOYEESETDNO=‘5‘,SUPERSSN=NULLWHERESSN=‘RNSCSE06‘;

UPDATEEMPLOYEESETDNO=‘1‘,SUPERSSN=‘RNSACC02‘WHERESSN=‘RNSACC01‘;

UPDATEEMPLOYEESETDNO=‘1‘,SUPERSSN=NULLWHERESSN=‘RNSACC02‘;

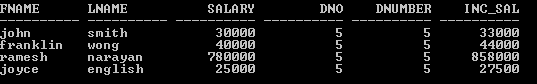
UPDATEEMPLOYEESETDNO=‘4‘,SUPERSSN=NULLWHERESSN=‘RNSISE01‘;

UPDATEEMPLOYEESETDNO=‘2‘,SUPERSSN=NULLWHERESSN=‘RNSIT01‘;

# Howtheresultingsalariesifeveryemployeeworkingonthe‘Research’Departmentsis givena10percentraise.

SQL>SELECTE.FNAME,E.LNAME,1.1\*E.SALARYASINCR\_SAL2FROMEMPLOYEE1 E,DEPARTMENTD,EMPLOYEE1W

1. WHEREE.SSN=W.SSN
2. ANDE.DNO=D.DNUMBER
3. ANDD.DNAME='research';



# Findthesumofthesalariesofallemployeesofthe‘Accounts’department,aswellas the maximum salary, the minimum salary, and the average salary in thisdepartment

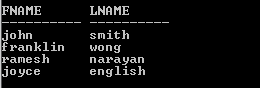
SQL> SELECT SUM(E.SALARY),MAX(E.SALARY),MIN(E.SALARY),AVG(E.SALARY)FROM EMPLOYEE1 E,DEPARTMENT D WHEREE.DNO=D.DNUMBERAND D.DNAME='RESEARCH';



# RetrievethenameofeachemployeeControlledbydepartmentnumber5(useEXISTSoperator).

SQL>SELECTE.FNAME,E.LNAME2FROMEMPLOYEE1E

3 WHEREEXISTS(SELECTDNOFROMEMPLOYEE1WHEREE.DNO=5);



# Retrievethenameofeach dept andnumberofemployeesworking ineachdepartmentwhichhas atleast2employees

SELECTDNAME, COUNT(\*)

FROMEMPLOYEEE,DEPARTMENTDWHERED.DNO=E.DNO

AND D.DNO IN (SELECT E1.DNOFROMEMPLOYEEE1

GROUP BY E1.DNO

having count(\*)>2 )ORDERBYDNO;

# Retrievethenameofemployeeswho bornin theyear1990’s

SELECTE.FNAME,E.LNAME,E.BDATEFROMEMPLOYEE1EWHEREBDATELIKE'196%';



# Retrievethenameofemployeesand theirdeptname(using JOIN)

SELECTE.FNAME,E.LNAME,DNAME

FROMEMPLOYEE ENATURALJOINDEPARTMENT DONE,DNO=D.DNO;

# Experiment5C

**PerformtheStringFunctions,DatefunctionsandMathematicalfunctionssupportedbyOracle**

SQL>selectascii('t')fromdual;ASCII('T')

116

SQL> select ascii('a') from dual;ASCII('A')

97

SQL> select ascii('A') from dual;ASCII('A')

65

SQL> select ascii('Z') from dual;ASCII('Z')

90

SQL>selectascii('z')fromdual;ASCII('Z')

122

SQL> SELECT UPPER('bldea sb arts and kcp science college') from dual;UPPER('BLDEASBARTSANDKCPSCIENCECOLLEG

BLDEASBARTSANDKCPSCIENCECOLLEGE

SQL>selectLOWER('welcometodbmslab')fromdual;LOWER('WELCOMETODBM

welcometodbmslab

SQL>selectLOWER('WELCOMETODBMSLAB')fromdual;

LOWER('WELCOMETODB

welcometodbmslab

SQL>SELECTREPLACE('HELLO','H','K')FROMDUAL;REPLA

KELLO

SQL> SELECT REPLACE('COMPUTER','C','K') FROM DUAL;REPLACE(

KOMPUTER

SQL>SELECTREPLACE('HELLO','L','A')FROMDUAL;REPLA

HEAAO

SQL>SELECTTRIM('A'FROM'ANACONDA')FROMDUAL;TRIM('

--

NACOND

SQL>SELECTLTRIM('ANACONDA','A')FROMDUAL;LTRIM('

NACONDA

SQL>SELECTLTRIM('ANIL','A')FROMDUAL;LTR

---NIL

SQL>SELECTRTRIM('ANITA','A')FROMDUAL;RTRI

ANIT

SQL>SELECTRTRIM('ANACONDA','A')FROMDUAL;RTRIM('

ANACOND

SQL>SELECTRTRIM('ANACONDA','A')FROMDUAL;RTRIM('ANAC

ANACONDA

**DateFunctions**

SQL>SELECTCURRENT\_DATEFROMDUAL;

CURRENT\_D

14-AUG-19

SQL> SELECT EXTRACT(YEAR FROM SYSDATE) FROM DUAL;EXTRACT(YEARFROMSYSDATE)

2019

SQL>SELECTEXTRACT(DAYFROMSYSDATE)FROMDUAL;EXTRACT(DAYFROMSYSDATE)

14

SQL>SELECTEXTRACT(MONTHFROMSYSDATE)FROMDUAL;EXTRACT(MONTHFROMSYSDATE)

8

SQL>SELECTSYSDATEFROMDUAL;SYSDATE

14-AUG-19

**MathematicalFunctions**

SQL>selectABS(-100)fromdual;ABS(-100)

100

SQL>selectABS(-6)fromdual;

ABS(-6)

6

SQL>selectFLOOR(2345.78)FROMDUAL;FLOOR(2345.78)

2345

SQL>SELECTGREATEST(23,67,90,123,78,50)FROMDUAL;GREATEST(23,67,90,123,78,50)

123

SQL>SELECTLEAST(34,21,67,11,89,9)FROMDUAL;LEAST(34,21,67,11,89,9)

9

SQL>SELECTLENGTH('RAJESHWARI')FROMDUAL;LENGTH('RAJESHWARI')

10

SQL>SELECTLENGTH(17245637)FROMDUAL;LENGTH(17245637)

8

SQL>SELECTSQRT(16)FROMDUAL;SQRT(16)

4

SQL>SELECTSQRT(99)FROMDUAL;SQRT(99)

9.94987437

SQL>SELECTPOWER(2,4)FROMDUAL;POWER(2,4)

16

SQL>SELECTPOWER(2,10)FROMDUAL;POWER(2,10)

1024

SQL>SELECTpower(2,10)FROMDUAL;POWER(2,10)

1024

SQL>SELECTROUND(5.86)FROMDUAL;ROUND(5.86)

6

SQL>SELECTROUND(1001.6)FROMDUAL;ROUND(1001.6)

1002

SQL>SELECTROUND(1001.3)FROMDUAL;ROUND(1001.3)

1001

SQL>SELECTSIN(90)FROMDUAL;SIN(90)

.893996664

SQL>SELECTCOS(45)FROMDUAL;COS(45)

.525321989

SQL>SELECTTAN(30)FROMDUAL;TAN(30)

-6.4053312

SQL>SELECTTAN(90)FROMDUAL;TAN(90)

-1.9952004

SQL>SELECTTAN(180)FROMDUAL;TAN(180)

1.33869021

SQL> SELECT SIGN(-128) FROM DUAL;SIGN(-128)

-1

SQL>SELECTSIGN(10)FROMDUAL;SIGN(10)

1

SQL>SELECTSIGN(0)FROMDUAL;SIGN(0)

0

SQL>SELECTLN(100)FROMDUAL;LN(100)

4.60517019

SQL>SELECTLN(10)FROMDUAL;LN(10)

2.30258509

SQL>SELECTLOG(10,100)FROMDUAL;LOG(10,100)

2

SQL>SELECTLOG(100,10)FROMDUAL;LOG(100,10)

.5

SQL>SELECTMOD(4,3)FROMDUAL;MOD(4,3)

1

SQL>SELECTMOD(4,2)FROMDUAL;

MOD(4,2)

0

SQL>SELECTEXP(2)FROMDUAL;

EXP(2)

7.3890561

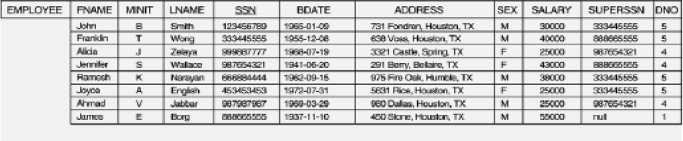
SQL> SELECT EXP(-2) FROM DUAL;EXP(-2)

.135335283

SQL>SELECTEXP(0)FROMDUAL;EXP(0)

1

# Experiment6

**Fora givenEMPLOYEEtables**

PerformtheFollowing

1. CreatingViews(WithandWithoutCheckOption),
2. SelectingfromaView
3. DroppingViews,

# SOLUTION:

SQL>CREATE TABLEEMPLOYEE(

SSNVARCHAR2(20)PRIMARYKEY,FNAMEVARCHAR2(20),

LNAMEVARCHAR2(20),

ADDRESSVARCHAR2(20),

SEX CHAR (1),SALARYINTEGER,

SUPERSSNREFERENCESEMPLOYEE(SSN),DNOREFERENCESDEPARTMENT(DNO));

SQL>DESCEMPLOYEE;

|  |  |  |
| --- | --- | --- |
| Name | Null? | Type |
| SSN | NOTNULL | VARCHAR2(20) |
| FNAME |  | VARCHAR2(20) |
| LNAME |  | VARCHAR2(20) |
| ADDRESS |  | VARCHAR2(20) |
| SEX |  | CHAR(1) |
| SALARY |  | NUMBER(38) |
| SUPERSSN |  | VARCHAR2(20) |
| DNO |  | NUMBER(38) |

INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES(‘RNSECE01‘,‘JOHN‘,‘SCOTT‘,‘BANGALORE‘,‘M‘,450000);

INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES(‘RNSCSE01‘,‘JAMES‘,‘SMITH‘,‘BANGALORE‘,‘M‘,500000);

INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES (‘RNSCSE02‘,‘HEARN‘,‘BAKER‘,‘BANGALORE‘,‘M‘, 700000);INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES(‘RNSCSE03‘,‘EDWARD‘,‘SCOTT‘,‘MYSORE‘,‘M‘,500000);

INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES (‘RNSCSE04‘,‘PAVAN‘,‘HEGDE‘,‘MANGALORE‘,‘M‘, 650000);INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES (‘RNSCSE05‘,‘GIRISH‘,‘MALYA‘,‘MYSORE‘,‘M‘,450000);

INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES (‘RNSCSE06‘,‘NEHA‘,‘SN‘,‘BANGALORE‘,‘F‘,800000);

INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES (‘RNSACC01‘,‘AHANA‘,‘K‘,‘MANGALORE‘,‘F‘,350000);

INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES (‘RNSACC02‘,‘SANTHOSH‘,‘KUMAR‘,‘MANGALORE‘,‘M‘, 300000);INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES (‘RNSISE01‘,‘VEENA‘,‘M‘,‘MYSORE‘,‘M‘,600000);

INSERTINTOEMPLOYEE(SSN,FNAME,LNAME,ADDRESS,SEX,SALARY)VALUES (‘RNSIT01‘,‘NAGESH‘,‘HR‘,‘BANGALORE‘,‘M‘,500000);

# CreatingView

Thequerythatdefinesthesales\_staffviewreferencesonlyrowsindepartment5.Furthermore,theCHECKOPTIONcreatestheviewwiththeconstraint(namedsales\_staff\_cnst) that INSERT and UPDATE statements issued against the view cannotresultinrowsthattheviewcannotselect.

# Creating Views(WithandWithoutCheckOption)

SQL>CREATEVIEWsales\_staffAS

* 1. SELECTfname,ssn,dno
  2. FROMemployee
  3. WHERE dno=5
  4. WITHCHECKOPTIONCONSTRAINTsales\_staff\_cnst;View created.

# Selecting froma View

SQL>select\*fromsales\_staff;

# DropView

SQL>DROPVIEWsales\_staff;

# Experiment7

WriteaPl/SQLprogramtoprintintegersfrom1to 10byusing PL/SQLFORloop

# SOLUTION:

**PL/SQLBlock**

SET SERVEROUTPUT ON SIZE 1000000;DECLARE

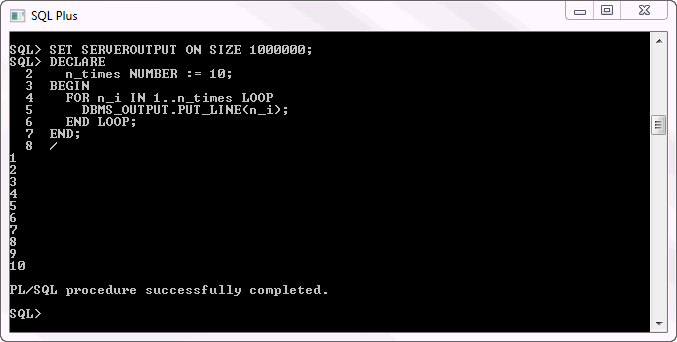
n\_timesNUMBER:=10;BEGIN

FOR n\_i IN 1..n\_times LOOPDBMS\_OUTPUT.PUT\_LINE(n\_i);ENDLOOP;

END;

/

# OutputTable



**Experiment8**

GiventhetableEMPLOYEE(EmpNo,Name,Salary,Designation,DeptID)writeacursortoselectthefivehighestpaidemployeesfromthetable.

EMPLOYEE(EmpNo,Name,Salary,Designation,DeptID)

# SOLUTION:

CREATETABLEEMPLOYEE(EMPNOINTEGERPRIMARYKEY,NAMEVARCHAR(20),

SALARY NUMBER(7,2),DESIGNATIONVARCHAR(10),DEPTIDINTEGER);

gete:/p8.sql;1declare

1. inumber:=0;
2. cursor ecisselectempno,name,salaryfromemployeeorder bygross\_salarydesc;4rec%rowtype;
3. begin
4. openec;
5. loop
6. exit when i=5;9fetchec into r;

10 dbms\_output.put\_line(r.emp\_no||' '||r.employee\_name||' '||r.salary);11i:=i+1;

1. endloop;
2. close ec;14\*end;15.

SQL>/

1. rajesh31000
2. paramesh15000
3. pushpa14000
4. vijaya14000
5. keerthi13000

PL/SQLproceduresuccessfullycompleted.

# Experiment10

Givenanintegeri, writeaPL/SQLproceduretoinsertthetuple(i,'xxx')intoagivenrelation.

# SOLUTION:

CREATE TABLE T2 (aINTEGER,

bCHAR(10));

CREATEORREPLACEPROCEDUREaddtuple1(xINNUMBER)AS

BEGIN

INSERTINTOT2VALUES(x,'xxx');

ENDaddtuple1;

.

run;

# VivaQuestions

1. **WhatisSQL?**

StructuredQueryLanguage

# Whatisdatabase?

A database is a logically coherent collection of data with some inherent meaning,representingsomeaspectofrealworldandwhichisdesigned, builtandpopulatedwith

data foraspecificpurpose.

# WhatisDBMS?

Itisacollectionofprogramsthatenablesuser tocreateandmaintainadatabase. Inotherwordsitisgeneral-purpose softwarethat providesthe userswiththe processesof

defining, constructingand manipulatingthedatabase forvariousapplications.

# WhatisaDatabasesystem?

ThedatabaseandDBMSsoftwaretogetheriscalledasDatabasesystem.

# AdvantagesofDBMS?

Redundancy is controlled.Unauthorized access is restricted.Providingmultipleuserinterfaces.Enforcingintegrityconstraints.

Providingbackupandrecovery.

# DisadvantageinFileProcessingSystem?

Dataredundancy&inconsistency.Difficultinaccessingdata.

Data isolation.Dataintegrity.

Concurrentaccessisnotpossible.SecurityProblems.

# Describethethreelevelsofdataabstraction?

Therearethreelevelsofabstraction:

Physicallevel:Thelowestlevelofabstractiondescribes howdataarestored.

Logicallevel*:* Thenexthigherlevelofabstraction, describeswhatdataarestoredindatabase andwhatrelationshipamongthose data.

Viewlevel:Thehighestlevelofabstractiondescribesonlypartofentiredatabase.

# Definethe"integrityrules"

TherearetwoIntegrityrules.

EntityIntegrity:Statesthat ―Primarykeycannot haveNULLvalue‖

ReferentialIntegrity:Statesthat―ForeignKeycanbeeitheraNULLvalueorshouldbePrimaryKeyvalueofotherrelation.

# What isextension andintension?

Extension - It is the number of tuples present in a table at any instance. This is time dependent.Intension-Itisaconstantvaluethatgivesthename, structureoftableandtheconstraintslaidonit.

# WhatisDataIndependence?

Data independence means that ―the application is independent of the storage structure andaccessstrategyofdata‖.Inother words,Theabilitytomodifytheschemadefinition inonelevelshouldnotaffectthe schema definitioninthe nexthigherlevel.

TwotypesofData Independence:

PhysicalDataIndependence:Modificationinphysicallevelshouldnotaffectthelogicallevel.LogicalDataIndependence:Modificationinlogical levelshouldaffecttheviewlevel.

NOTE:LogicalDataIndependenceismore difficultto achieve

# What isaview?Howitisrelatedto dataindependence?

A view may be thought of as a virtual table, that is, a table that does not really exist in its ownright but is instead derived from one or more underlying base table. In other words, there is nostoredfilethatdirectrepresentstheviewinsteadadefinitionofviewisstoredindatadictionary.Growth and restructuring of base tables is not reflected in views. Thus the view can insulateusers from the effects of restructuring and growth in the database. Hence accounts for logicaldata independence.

# WhatisDataModel?

Acollectionofconceptualtoolsfor describingdata,datarelationshipsdatasemanticsandconstraints.

# WhatisE-Rmodel?

This data model is based on real world that consists of basic objects called entities and ofrelationshipamongtheseobjects.Entitiesaredescribedinadatabasebyasetofattributes.

# WhatisObjectOrientedmodel?

This model is based on collection of objects. An object contains values stored in instancevariables within the object. An object also contains bodies of code that operate on the object.Thesebodiesofcodearecalledmethods. Objectsthatcontainsametypesofvaluesandthesamemethodsaregrouped togetherintoclasses.

# What isanEntity?

Itisan'object'intherealworldwithanindependentexistence.

# What isanEntitytype?

Itisacollection(set)ofentitiesthathavesameattributes.

# What isanEntity set?

It isacollectionofallentitiesofparticularentitytypeinthe database.

# What isanExtensionofentitytype?

Thecollectionsofentitiesofaparticularentitytype aregroupedtogetherinto anentityset.

# Whatisanattribute?

Itisaparticularproperty, whichdescribestheentity.

# What isaRelation Schemaanda Relation?

A relation Schema denoted by R(A1, A2, …, An) is made up of the relation name R and the listofattributesAithatitcontains.Arelationisdefinedasasetoftuples.Letrbetherelationwhichcontainssettuples(t1,t2,t3,...,tn).Eachtupleisanorderedlistofn-valuest=(v1,v2,...,vn).

# What isdegreeofaRelation?

Itisthenumber ofattributeofitsrelationschema.

# What isRelationship?

Itisanassociationamong twoormoreentities.

# WhatisRelationshipset?

Thecollection(or set)ofsimilar relationships.

1. ***WhatisRelationshiptype?***

Relationshiptypedefinesaset ofassociationsorarelationshipsetamong agivensetofentitytypes.

# WhatisdegreeofRelationship type?

Itisthenumberofentitytypeparticipating.

# WhatisDDL(Data DefinitionLanguage)?

AdatabaseschemaisspecifiedbyasetofdefinitionsexpressedbyaspeciallanguagecalledDDL.

# What isVDL(ViewDefinition Language)?

Itspecifiesuserviewsandtheirmappingstotheconceptualschema.

# What isSDL(StorageDefinition Language)?

Thislanguageistospecifytheinternalschema. This languagemayspecifythemappingbetweentwoschemas.

# WhatisDataStorage -DefinitionLanguage?

The storage structures and access methods used by database system are specified by a set ofdefinitionina specialtype ofDDLcalleddata storage-definitionlanguage.

# WhatisDML(DataManipulationLanguage)?

This languagethatenableuser toaccessormanipulatedataasorganizedbyappropriatedatamodel.

ProceduralDMLorLowlevel:DMLrequiresauserto specifywhatdataareneeded andhowtogetthosedata.

Non-ProceduralDMLorHighlevel:DMLrequiresausertospecifywhatdataareneededwithoutspecifyinghowtogetthosedata.

# WhatisDMLCompiler?

It translates DML statements in a query language into low-level instruction that the queryevaluationenginecanunderstand.

# WhatisRelationalAlgebra?

It isa proceduralquerylanguage.Itconsistsofa set ofoperationsthat takeoneortworelationsasinputandproduce anewrelation.

# WhatisRelationalCalculus?

Itisanapplied predicatecalculusspecificallytailoredforrelationaldatabasesproposed byE.F.Codd.E.g.oflanguagesbasedonitare DSL,ALPHA,QUEL.

# Whatisnormalization?

Itisaprocessofanalyzing thegivenrelationschemasbased ontheirFunctionalDependencies(FDs)andprimarykeytoachievetheproperties

Minimizingredundancy

Minimizinginsertion,deletionandupdateanomalies.

# WhatisFunctionalDependency?

A Functional dependency is denoted by X Y between two sets of attributes X and Y that aresubsetsofRspecifiesa constraintonthepossible tuplethatcanforma relationstaterofR.Theconstraint is for any two tuples t1 and t2 in r if t1[X] = t2[X] then they have t1[Y] = t2[Y]. Thismeansthe value ofX componentofa tuple uniquelydeterminesthe value ofcomponentY.

# WhenisafunctionaldependencyFsaidtobeminimal?

EverydependencyinFhasasingleattributeforitsrighthandside.

Wecannotreplace anydependencyX AinFwitha dependencyYAwhere Yisa propersubsetofXandstillhave asetofdependencythatisequivalenttoF.

Wecannot removeanydependencyfromFandstillhaveset ofdependencythatisequivalenttoF.

# WhatisMultivalueddependency?

Multivalued dependency denoted by X Y specified on relation schema R, where X and Y arebothsubsetsofR,specifiesthe followingconstraintonanyrelationrofR:iftwotuplest1andt2exist in r such that t1[X] = t2[X] then t3 and t4 should also exist in r with the followingproperties

t3[x]=t4[X]=t1[X] =t2[X]

t3[Y]=t1[Y]andt4[Y]=t2[Y]

t3[Z]=t2[Z]andt4[Z] =t1[Z]

where [Z=(R-(X UY))]

# WhatisLosslessjoinproperty?

Itguaranteesthat thespurioustuplegenerationdoesnotoccurwithrespect torelationschemasafterdecomposition.

# What is1NF(NormalForm)?

Thedomainofattribute mustincludeonlyatomic(simple,indivisible)values.

# WhatisFullyFunctionaldependency?

It is based on concept of full functional dependency. A functional dependency X Y is fullyfunctionaldependencyifremovalofanyattributeA fromXmeansthatthe dependencydoes notholdanymore.

# What is2NF?

ArelationschemaRisin2NF if itisin1NFandeverynon-primeattributeAinRisfullyfunctionallydependentonprimarykey.

# What is3NF?

ArelationschemaRisin3NF if itisin2NFandforeveryFDXAeitherofthefollowingistrueX is aSuper-keyofR.

Aisaprime attributeofR.

Inotherwords,ifeverynonprimeattributeisnon-transitivelydependent onprimarykey.

# What isBCNF(Boyce-Codd NormalForm)?

A relation schema R is in BCNF if it is in 3NF and satisfies additional constraints that for everyFD XA,Xmustbeacandidate key.

# What is4NF?

ArelationschemaRissaidtobein4NFif for everyMultivalueddependencyXYthatholdsoverR,oneoffollowingis true

X issubsetorequalto(or)XY =R.X isasuperkey.

# What is5NF?

ARelationschemaRissaidtobe5NFiffor everyjoindependency{R1, R2,...,Rn}thatholdsR,onethefollowingis true

Ri=Rforsomei.

Thejoindependencyisimplied bytheset ofFD,overRinwhichtheleftsideiskeyofR.