

Karnatak Law Society's  
**GOGTE INSTITUTE OF TECHNOLOGY**  
 Department of Computer Science and Engineering  
 Udyambag Belgavi – 590008

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)  
**(APPROVED BY AICTE, NEW DELHI)**



Journal Submission of  
**DATABASE APPLICATION LABORATORY (18CSL48)**  
 on  
**DATABASE MANAGEMENT SYSTEM**

*Submitted in the partial fulfilment for the academic requirement of*  
**4<sup>th</sup> semester B.E.**  
**in**  
**Computer Science Engineering**

Submitted by

NAME	USN
VENKATESH G D	2GI19CS175

Under the guidance of  
 Prof. Kuldeep Sambrekar  
 Asst. Prof., Dept. of CSE  
 Academic Year 2021 (Even Semester)

## DEPARTMENT OF COMPUTER ENGINEERING



### **CERTIFICATE**

This is to certify that the Journal Term works carried out by **Venkatesh.**

**G. D** bearing **USN: 2GI19CS175** has submitted in partial fulfilment of the requirements for 4<sup>th</sup> semester B.E. in **Database Management System, COMPUTER SCIENCE AND ENGINEERING**, Visvesvaraya Technological University, Belagavi. It is certified that all corrections/suggestions indicated have been incorporated in the journal. The journal has been approved as it satisfies the academic requirements in respect of research work prescribed for the said degree.

Date: 15/07/2021

Signature of Guide

Place: Belagavi

**Prof. Kuldeep Sambrekar**

Asst. Prof., Dept. of CSE,

KLS Gogte Institute of Technology, Belagavi

Name of the Examiners

Signature of the Examiners

1. \_\_\_\_\_

1. \_\_\_\_\_

2. \_\_\_\_\_

2. \_\_\_\_\_

## INDEX

<b>TERMWORK NO.</b>	<b>TITLE</b>
1	National Hockey League
2	An Educational Institute
3	Airline Flight Database
4	Order Database
5	Movie Database

## **Design of ER-Model for National Hockey League**

Expt no: 1

Date: 26-05-2021

### **Problem Statement:**

Suppose you are given the following requirements for a simple database for the National Hockey League (NHL):

- the NHL has many teams,
- each team has a name, a city, a coach, a captain, and a set of players,
- each player belongs to only one team,
- each player has a name, a position (such as left wing or goalie), a skill level, and a set of injury records,
- a team captain is also a player,
- a game is played between two teams (referred to as host\_team and guest\_team) and has a date (such as May 11th, 1999) and a score (such as 4 to 2).

Design an ER-Model for this application scenario using all the standard notations of ER-Model. Apply the ER-to-Relational Rules and normalization to get the relational schema and do the following:

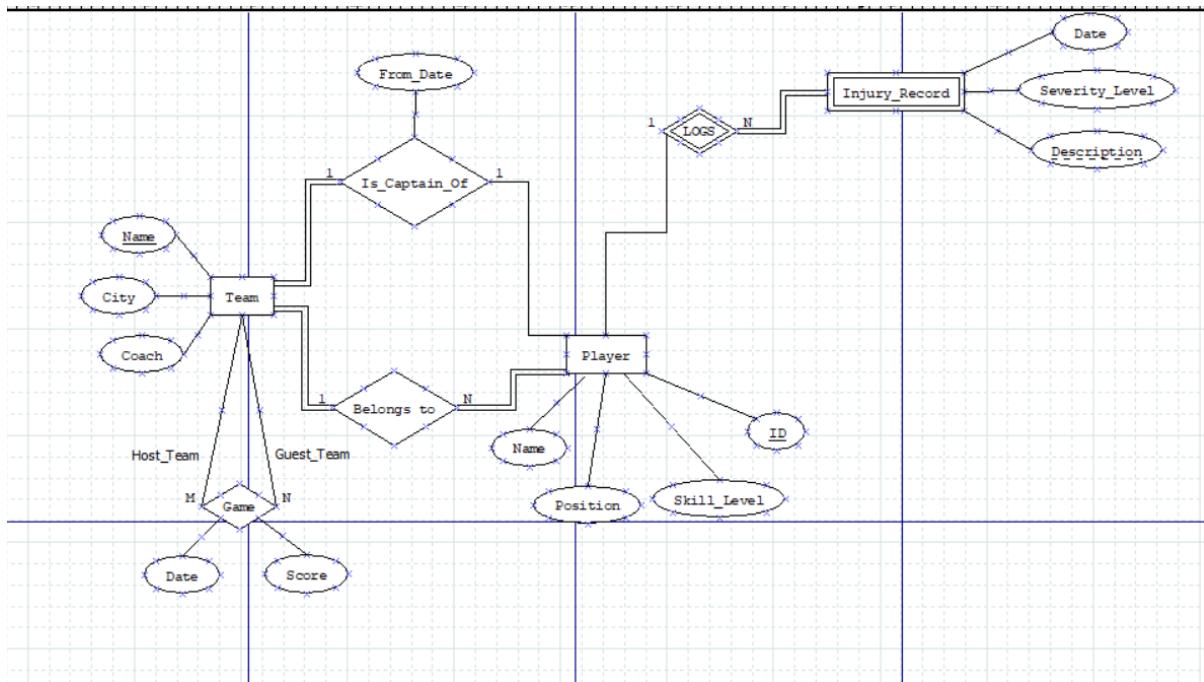
- a. Create the database with all necessary constraints (Primary and Foreign keys)
- b. Populate each table with appropriate data
- c. Execute queries on the tables created. (Open ended)
- d. Create graphical user interfaces (GUI) using HTML/PHP/VB.Net/Java

Concepts: ER-Modelling is a first step towards building a database application. It helps in identifying various entities, their attributes and the relationship between them. The ER-Model helps the application developers to explain the customers, what all data would be stored and seek their suggestions to include all the data relevant to the application. While designing an ER-Model it is important to include only the attributes relevant to the entity types. Further after drawing ER-diagram the structural constraints, namely the cardinality ratios and participation constraints must be correctly indicated. There 7 mapping rules, which must be applied to the ER-diagram after completion to get the Relational model. The relational schema diagram for each relation must be drawn and the Primary key and Foreign keys must be correctly indicated. Then the DDL statements must be used to create the tables in Oracle DBMS. Using INSERT command data must be inserted and using SQL queries the data must be checked for its correctness.

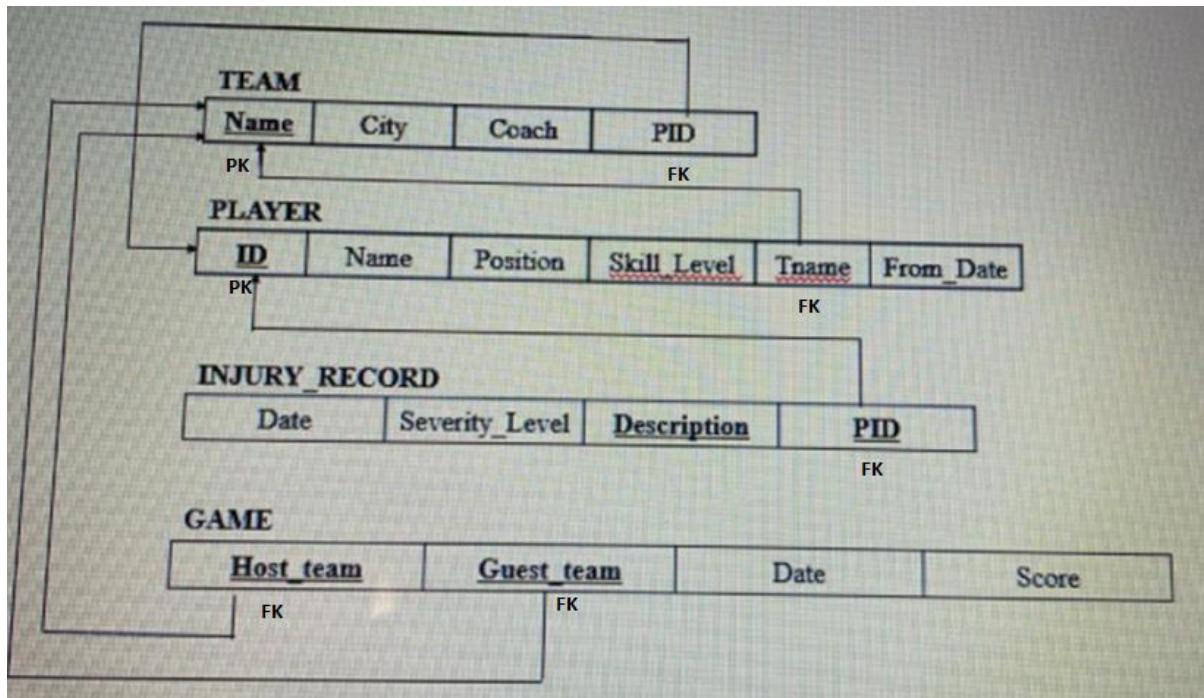
#### Learning Outcomes:

1. ER-model is a set of concepts to describe data in graphical form.
2. There are 7 ER-to-Relational mapping rules to get Relational model from ER-Model.
3. Relational model is a set of concepts to describe data to RDBMS.
4. Relation, tuple, attribute, domain, Primary key, foreign key are the concepts in Relational mode.
5. DDL statements help us to create tables and specify constraints.
6. DML statements help us to populate and manipulate the database.
7. Design a GUI and write the program to connect to DBMS Server and display the data.
8. Learned to use SQL queries to list data stored in tables.

### ER Diagram:



### Schema Diagram:



Create the Database with all the necessary constraints:

Create the database with all necessary constraints:

1. TABLE TEAM

```
SQL> CREATE TABLE TEAM
    CNAME VARCHAR(20),
    CITY VARCHAR(10),
    COACH VARCHAR(20) NOT NULL,
    CPID INT,
    PRIMARY KEY (CNAME));
```

Table created.

```
SQL> CREATE TABLE PLAYER
```

```
CID INT,
NAME VARCHAR(20) NOT NULL,
POSITION VARCHAR(10),
SKILL_LEVEL INT NOT NULL,
TNAME VARCHAR(20),
FROM_DATE DATE,
PRIMARY KEY (ID)
FOREIGN KEY (TNAME) REFERENCES TEAM(NAME);
```

Name	Null?	Type
ID	NOT NULL	NUMBER(38)
NAME	NOT NULL	VARCHAR2(20)
POSITION		VARCHAR2(10)
SKILL_LEVEL	NOT NULL	NUMBER(38)
TNAME		VARCHAR2(20)
FROM_DATE		DATE

SQL> ALTER TABLE TEAM ADD FOREIGN KEY (CPID)  
REFERENCES PLAYER (ID);

Name	Null?	Type
NAME	NOT NULL	VARCHAR2(20)
CITY		VARCHAR2(10)
COACH	NOT NULL	VARCHAR2(20)
CPID		NUMBER(38)

SQL> CREATE TABLE INJURY\_RECORD  
C ID DATE DATE,  
SEN\_LEVEL VARCHAR(10),  
DESCRIPTION VARCHAR(10),  
PID INT,  
PRIMARY KEY (DESCRIPTION, PID),  
FOREIGN KEY (PID) REFERENCES PLAYER (ID)  
ON DELETE CASCADE);

Table created

```
SQL> DESC INJURY_RECORD
      Name          Null?    Type
-----  -----
IDATE           DATE
SEV_LEVEL       VARCHAR2(10)
DESCRIPTION     NOT NULL VARCHAR2(10)
PID             NOT NULL NUMBER(38)
```

SQL> CREATE TABLE GAME
HOST\_TEAM VARCHAR(10),
GUEST\_TEAM VARCHAR(10),
GDATE DATE NOT NULL,
SCORE INT NOT NULL,
PRIMARY KEY (HOST\_TEAM) REFERENCES TEAM
(CNAME) ON DELETE SET NULL;

Table created

```
SQL> DESC GAME
      Name          Null?    Type
-----  -----
HOST_TEAM       NOT NULL VARCHAR2(20)
GUEST_TEAM      NOT NULL VARCHAR2(20)
GDATE          NOT NULL DATE
SCORE           NOT NULL NUMBER(38)
```

Insert statements for each table

SQL> INSERT INTO TEAM

VALUES ('&NAME', '&CITY', '&COACH', &CPID);

Enter value for name : Flying Elbows

Enter value for city : Haryana

Enter value for coach : Sommayya

old 1: INSERT INTO TEAM

VALUES ('&NAME', '&CITY', '&COACH', &CPID)

new 1: INSERT INTO TEAM

VALUES ('Flying Elbows', 'Haryana', 'Sommayya', NULL)

SQL> SELECT \* FROM TEAM;

NAME	CITY	COACH	CPID
Flying Elbows	HARIYANA	Sommayya	
Lights Out	GUJARAT	Pargat	
Zambronies	BILASPUR	Dilip Tirkey	
Goals Galore	PUNJAB	Bharat	
Stick Figures	COORG	P. R. Sreejesh	

```
SQL> INSERT INTO PLAYER
VALUES(&ID, '&NAME', '&POSITION', &SKILL-LEVEL,
      '&TNAME', '&FROM-DATE');
```

Enter value for id : 10

Enter value for name: Balbir Singh

Enter value for position: Forward.

Enter value for skill-level: 5

Enter value for tname : Lights Out

Enter value for from-date : 12-JAN-2000

```
old 1: INSERT INTO PLAYER
```

```
VALUES(&ID, '&NAME', '&POSITION', &SKILL-LEVEL,
      '&TNAME', '&FROM-DATE')
```

```
new 1: INSERT INTO PLAYER
```

```
VALUES(10, 'Balbir Singh', 'Forward', 5, 'Lights Out', '12-JAN-2000')
```

```
SQL> SET LINESIZE 150
SQL> SELECT * FROM PLAYER;
```

ID	NAME	POSITION	SKILL_LEVEL	TNAME	FROM_DATE
10	Balbir Singh	Forward	5	Lights Out	12-JAN-00
20	Leslie	Midfield	2	Flying Elbows	03-MAR-01
30	Charanjit	Fullbacks	3	Zambronies	05-FEB-15
40	Harmik	Sweeper	4	Stick Figures	23-JUL-08
50	Ajit Pal	Goalie	5	Goals Galore	09-DEC-17

SQL> UPDATE TEAM SET CPID=10 WHERE  
NAME = 'Flying Elbows';

i now updated.

SQL> UPDATE TEAM SET CPID=20 WHERE  
NAME = 'Lights Out';

i now updated.

SQL> UPDATE TEAM SET CPID=30 WHERE  
NAME = 'Zamboniies';

i now updated.

SQL> UPDATE TEAM SET CPID=40 WHERE  
NAME = 'Goals Galore';

i now updated

SQL> UPDATE TEAM SET CPID=50 WHERE  
NAME = 'Stick Figures';

i now updated

SQL> SELECT \* FROM TEAM;

NAME	CITY	COACH	CPID
Flying Elbows	HARIYANA	Sommayya	10
Lights Out	GUJARAT	Pargat	20
Zambronies	BILASPUR	Dilip Tirkey	30
Goals Galore	PUNJAB	Bharat	40
Stick Figures	COORG	P. R. Sreejesh	50

SQL> INSERT INTO GAME  
VALUES ('&HOST\_TEAM', '&GUEST\_TEAM', '&GDATE',  
& SCORE);

Enter value for host-team: Flying Elbows

Enter value for guest-team: Lights Out

Enter value for gdate: 12-JAN-00

Enter value for score: 3.

old 1: INSERT INTO GAME

VALUES ('&HOST\_TEAM', '&GUEST\_TEAM', '&GDATE',  
& SCORE);

new 1: INSERT INTO ~~INJURY.RECORD~~ GAME

VALUES ('Flying Elbows', 'Lights Out', '12-JAN-00', 3);

SQL> SELECT \* FROM INJURY\_RECORD;

IDATE	SEV_LEVEL	DESCRIPTION	PID
20-DEC-00	HIGH	HAND	10
12-JAN-99	MEDIUM	LEG	20
13-JUL-05	LOW	HEAD	10
15-SEP-02	LOW	LEG	30
17-NOV-02	HIGH	HEAD	40

SQL> INSERT INTO GAME  
VALUES ('&HOST\_TEAM', '&GUEST\_TEAM', '&GDATE',  
& SCORE);

Enter value for host-team: Flying Elbows

Enter value for guest-team: Lights Out

Enter value for gdate: 12-JAN-00

Enter value for score: 3.

old \$: INSERT INTO GAME  
VALUES ('&HOST\_TEAM', '&GUEST\_TEAM', '&GDATE',  
& SCORE);

new \$: INSERT INTO INJURY\_RECORD

VALUES ('Flying Elbows', 'Lights Out', '12-JAN-00', 3);

```
SQL> SELECT * FROM GAME;
```

HOST_TEAM	GUEST_TEAM	GDATE	SCORE
Flying Elbows	Lights Out	12-JAN-00	3
Lights Out	Zambronies	03-MAR-01	4
Zambronies	Flying Elbows	23-JUL-08	4
Goals Galore	Stick Figures	05-FEB-15	3
Goals Galore	Lights Out	09-DEC-17	3

Execute queries on the tables created. (Open ended)

1. List all the Players whose injury severity level is high and who's name starts with 'Ba'

```
SQL> SELECT NAME
  2  FROM PLAYER P, INJURY_RECORD I
  3 WHERE P.ID = I.PID AND SEV-LEVEL = 'HIGH' +
        NAME LIKE 'Ba%';
```

```
NAME
```

```
Balbir Singh
```

2. Count the Host teams those who have participated in more than one game.

```
SQL> SELECT HOST-TEAM, COUNT(HOST-TEAM)
  2  FROM GAME
  3  GROUP BY HOST-TEAM
  4  HAVING COUNT(*) > 1;
```

HOST_TEAM	COUNT(HOST_TEAM)
Goals Galore	2

3. List the names of the Coach in ascending order.

S Q L > SELECT NAME, COACH
2 FROM TEAM
3 ORDER BY COACH ASC ;

NAME	COACH
Goals Galore	Bharat
Zambronies	Dilip Tirkey
Stick Figures	P. R. Sreejesh
Lights Out	Pargat
Flying Elbows	Sommayya

4. List the names of the Players who started their career in Hockey league from year 2000 to 2010.

S Q L > SELECT ID, NAME, FROM_DATE
2 FROM PLAYER
3 WHERE FROM_DATE LIKE '_____0-';

ID	NAME	FROM_DATE
10	Balbir Singh	12-JAN-00
20	Leslie	03-MAR-01
40	Harmik	23-JUL-08

5. Count the total number of Game scores by grouping score.

```
SQL> SELECT SCORE, COUNT(*)
      2 FROM GAME
      3 GROUP BY SCORE;
```

SCORE	COUNT(*)
4	2
3	3

### Graphical User Interface:

### Outputs:

The screenshot shows a graphical user interface with a main window and a modal dialog box.

**Main Window:**

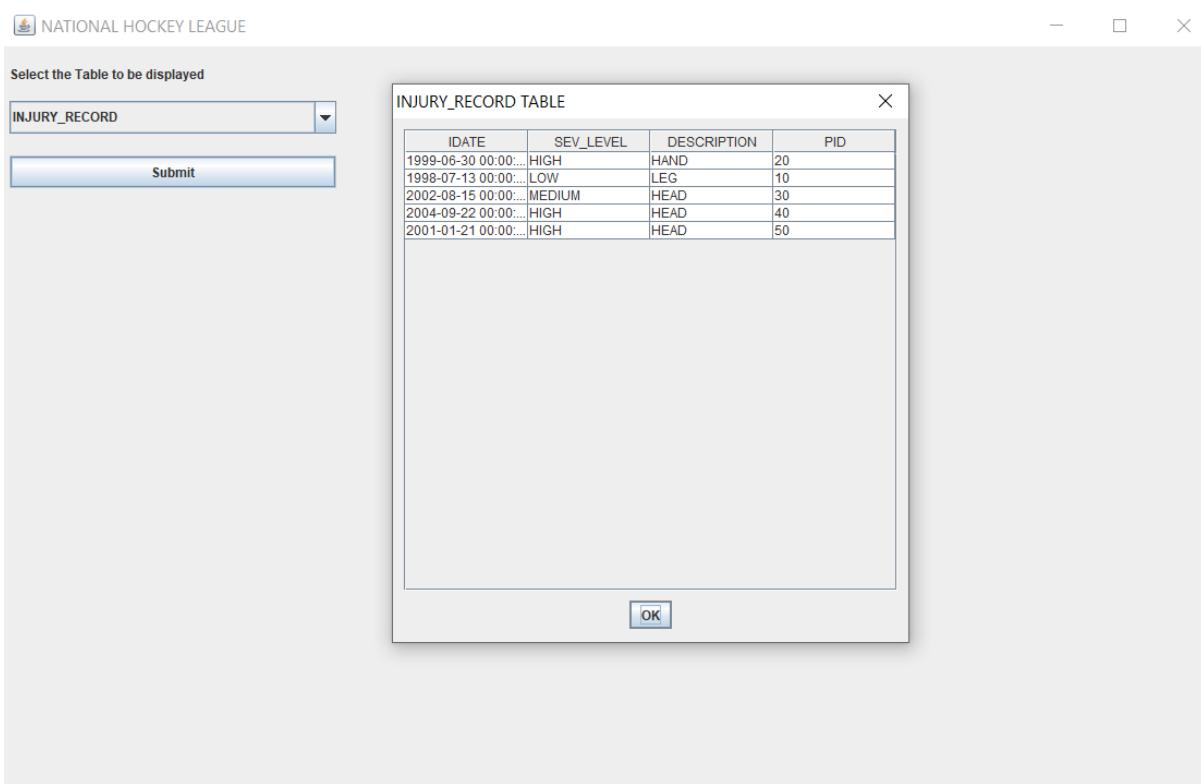
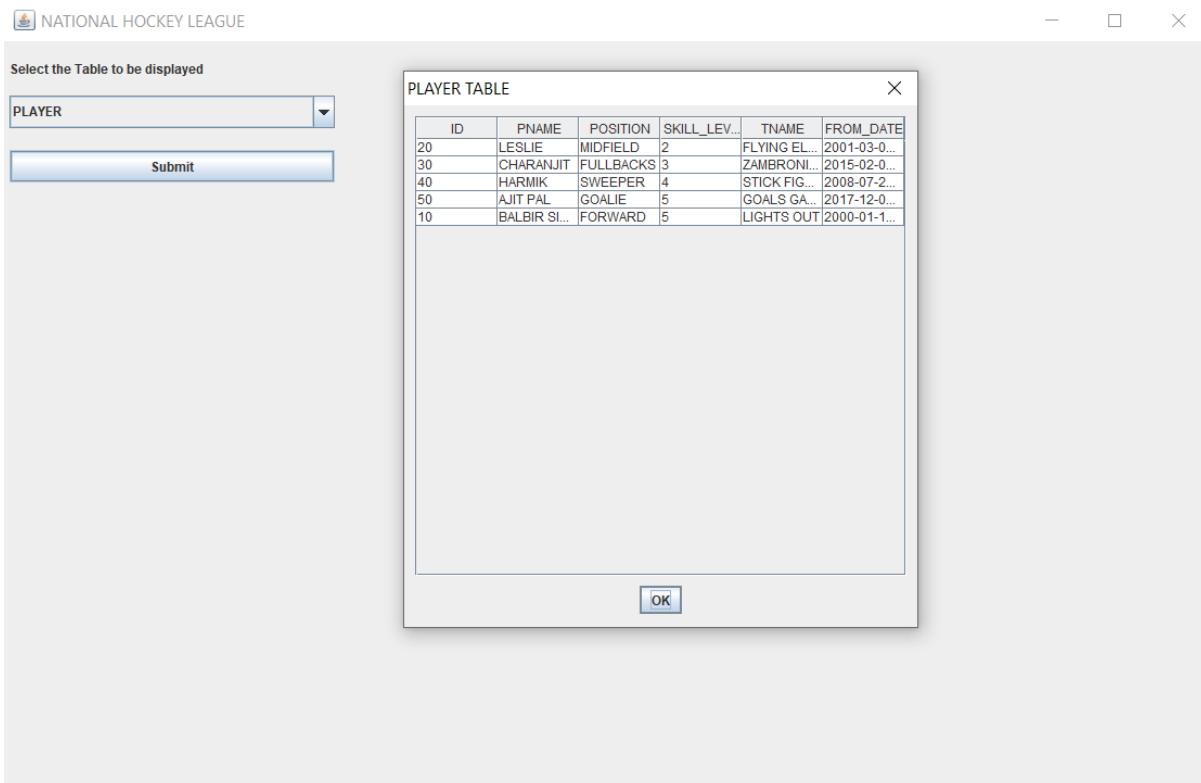
- Icon: National Hockey League logo
- Title: NATIONAL HOCKEY LEAGUE
- Text: Select the Table to be displayed
- Dropdown menu: TEAM
- Button: Submit

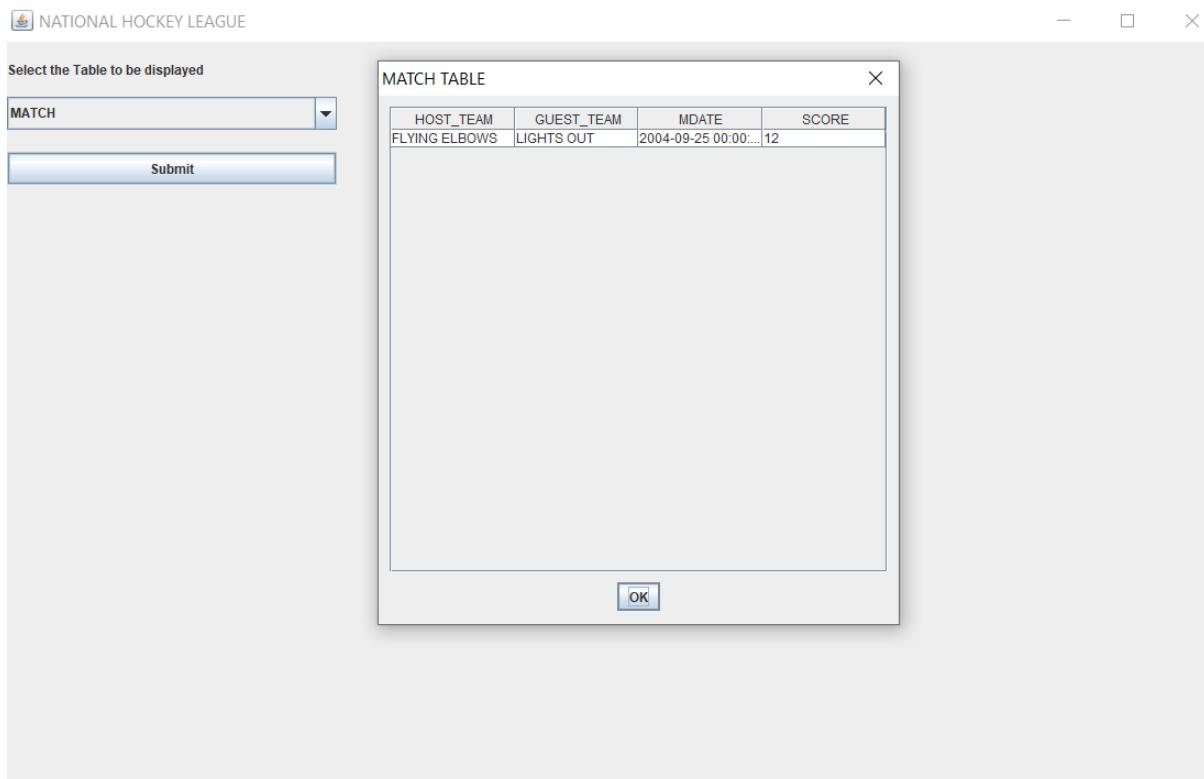
**Modal Dialog Box (TEAM TABLE):**

- Title: TEAM TABLE
- Table Data:

NAME	CITY	COACH	CPID
FLYING ELBOWS	HARIYANA	SOMMAYYA	10
LIGHTS OUT	GUJARAT	PARGAT	20
ZAMBONIES	BILASPUR	DILIP TIRKEY	30
GOALS GALORE	PUNJAB	BHARAT	40
STICK FIGURES	COORG	P.R.SREEJESH	50

- OK button at the bottom right of the dialog box





## Conclusion:

We learnt to use the open-source ER-design tool dia and created the ER-Model for the above said problem statement. Converted the ER-diagram into relational schema diagram by applying the ER- to relational rules. We identified primary and foreign keys and created all the relations in Oracle DBMS using DDL statements. Further, the database was populated with real data using insert statements. The content of each table was displayed using SELECT sql statement. We learnt about update and delete sql statements and also learnt about alter table command to modify or add constraints to the table structure after they are created. Learnt how a GUI can be built to connect to a back-end database in ORACLE / MYSQL using JAVA / PHP.

## **Design of ER-Model for an Educational Institute**

Expt no: 2

Date: 02-06-2021

### **Problem Statement:**

Design an ER-Model for an educational institute which is required to record the student's attendance and IA performance in all the subjects and inform the same to their parents. The institute will have many departments, each with its own faculty and Head of the department. The subjects the student's study can be either elective or core. A faculty has to take at least one subject and at most 2 subjects and the subjects are not shared. The students take 3 tests and the average is computed by taking average of best two of the three scores. The model be designed to record only the CIE marks and not SEE marks. After the ER-Model, map it to relational schema by identifying Primary and Foreign keys. Normalize and do the following.

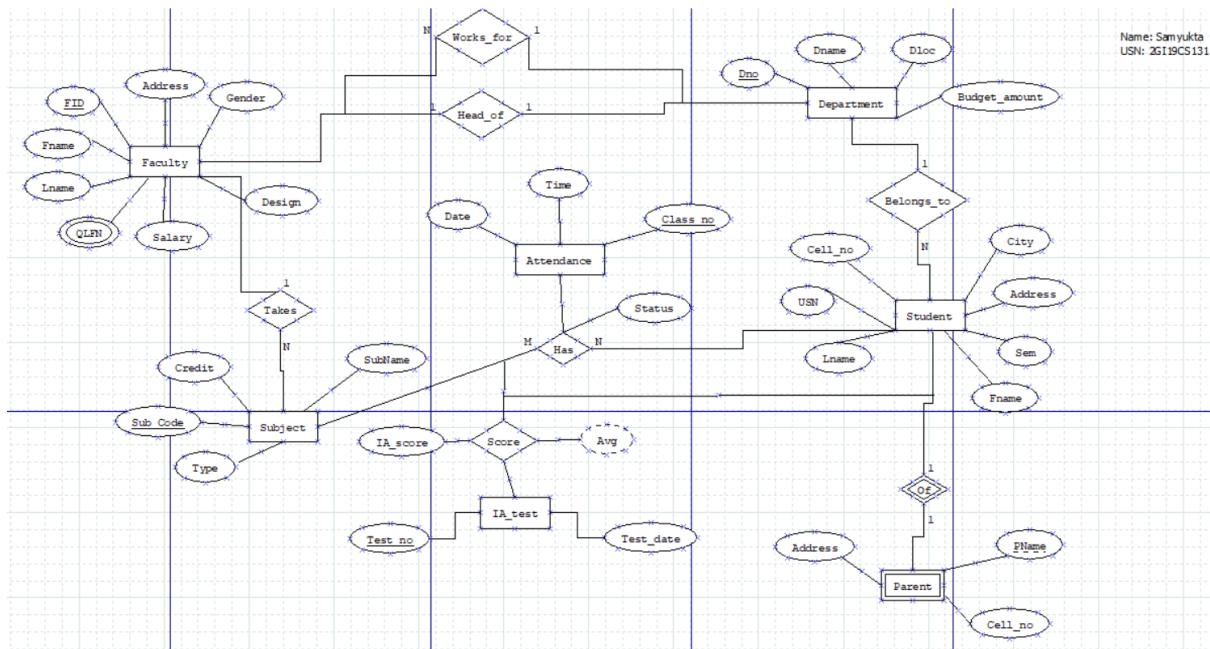
- a. Create the database with all necessary constraints  
(Primary and Foreign keys)
- b. Populate each table with appropriate data
- c. Execute queries on the tables created. (Open ended)
- d. Create graphical user interfaces (GUI) using  
HTML/PHP/VB.Net/Java

Concepts: ER-Modelling is a first step towards building a database application. It helps in identifying various entities, their attributes and the relationship between them. The ER-Model helps the application developers to explain the customers, what all data would be stored and seek their suggestions to include all the data relevant to the application. While designing an ER-Model it is important to include only the attributes relevant to the entity types. Further after drawing ER-diagram the structural constraints, namely the cardinality ratios and participation constraints must be correctly indicated. There 7 mapping rules, which must be applied to the ER-diagram after completion to get the Relational model. The relational schema diagram for each relation must be drawn and the Primary key and foreign keys must be correctly indicated. Then the DDL statements must be used to create the tables in Oracle DBMS. Using INSERT command data must be inserted and using SQL queries the data must be checked for its correctness.

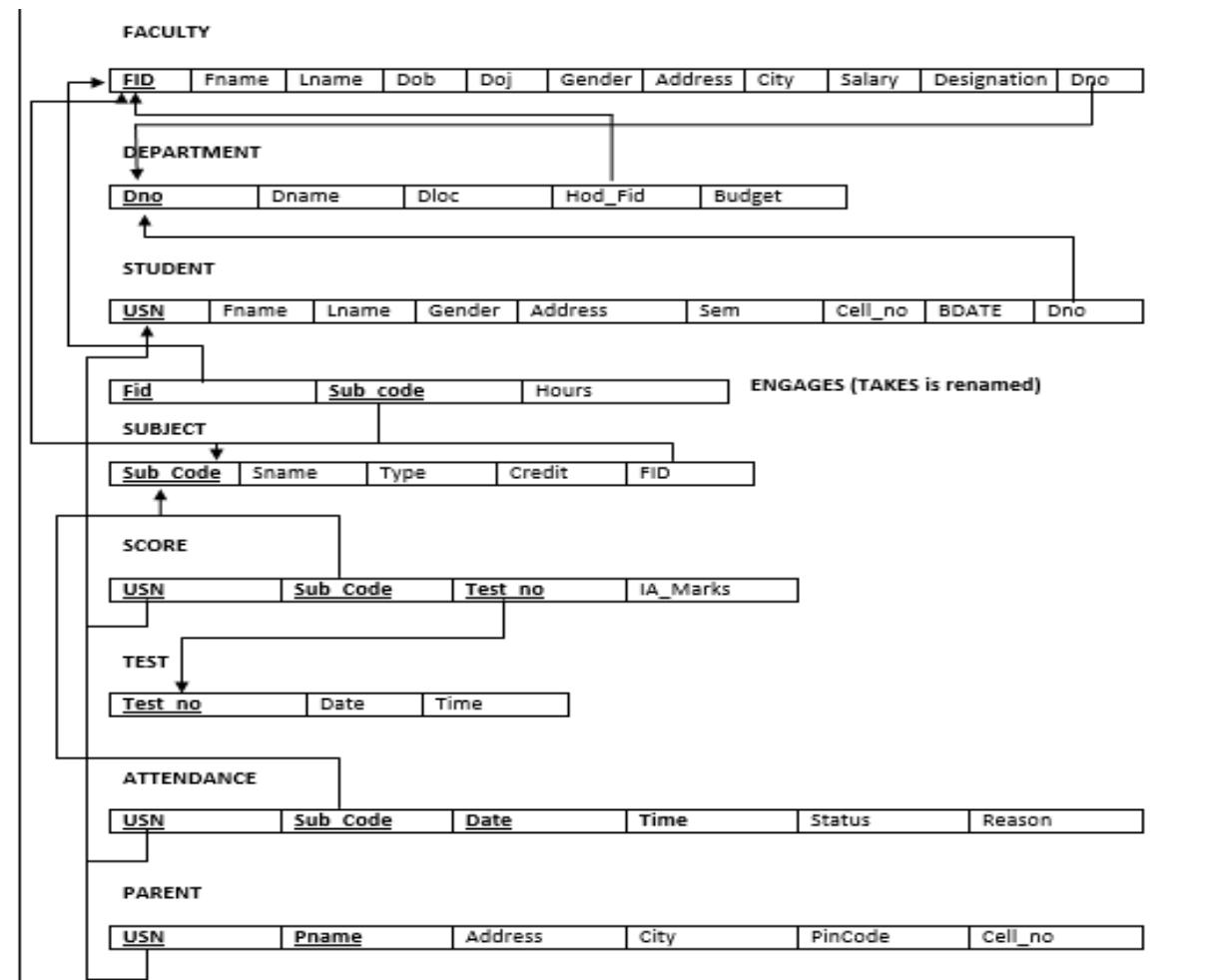
Learning Outcomes:

9. ER-model is a set of concepts to describe data in graphical form.
10. There are 7 ER-to-Relational mapping rules to get Relational model from ER-Model.
11. Relational model is a set of concepts to describe data to RDBMS.
12. Relation, tuple, attribute, domain, Primary key, foreign key are the concepts in Relational mode.
13. DDL statements help us to create tables and specify constraints.
14. DML statements help us to populate and manipulate the database.
15. Design a GUI and write the program to connect to DBMS Server and display the data.
16. Learned to use SQL queries to list data stored in tables.

## ER Diagram:



## Schema Diagram:



C. Create the database with all necessary constraints (Primary and Foreign keys)

Name	Null?	Type
FID	NOT NULL	NUMBER(38)
FNAME	NOT NULL	VARCHAR2(20)
LNAME		VARCHAR2(20)
DOB		DATE
DOJ		DATE
GENDER		CHAR(1)
ADDRESS		VARCHAR2(20)
CITY		VARCHAR2(20)
SALARY		NUMBER(38)
DISGNATION		VARCHAR2(20)
DNO		NUMBER(38)

## 2. TABLE DEPARTMENT

```
SQL> CREATE TABLE DEPARTMENT
      DNO INT,
      DNAME VARCHAR(20) NOT NULL,
      DLOC VARCHAR(20),
      HOD_FID INT,
      PRIMARY KEY(DNO),
      FOREIGN KEY(HOD_FID) REFERENCES FACULTY(FID);
```

Table created.

SQL> DESC DEPARTMENT

Name	Null?	Type
DNO	NOT NULL	NUMBER(38)
DNAME	NOT NULL	VARCHAR2(20)
DLOC		VARCHAR2(20)
HOD_FID		NUMBER(38)

SQL> ALTER TABLE FACULTY ADD FOREIGN KEY (DNO)  
REFERENCES DEPARTMENT (DNO);

Table altered.

### 3. TABLE STUDENT:

SQL> CREATE TABLE STUDENT

```
C USN VARCHAR (10),
    FNAME VARCHAR (20) NOT NULL,
    LNAME VARCHAR (20),
    GENDER CHAR (1),
    ADDRESS VARCHAR (10),
    SEM INT NOT NULL,
    MOB_NO INT
    BDATE DATE,
    DNO INT,
    PRIMARY KEY (CUSN),
    FOREIGN KEY (DNO) REFERENCES DEPARTMENT (DNO);
```

OTE 7 PRO  
CAMERA Table created.

```
SQL> DESC STUDENT
Name          Null?    Type
-----        -----
USN           NOT NULL VARCHAR2(10)
FNAME          NOT NULL VARCHAR2(20)
LNAME          VARCHAR2(20)
GENDER         CHAR(1)
ADDRESS         VARCHAR2(10)
SEM            NOT NULL NUMBER(38)
MOB_NO          NUMBER(38)
BDATE           DATE
DNO            NUMBER(38)
```

#### 4. TABLE SUBJECT :

```
SQL> CREATE TABLE SUBJECT
  (SUB_CODE VARCHAR(8),
   SNAME VARCHAR(20) NOT NULL,
   TYPE VARCHAR(10) NOT NULL,
   CREDIT INT,
   FID INT,
   PRIMARY KEY (SUB_CODE)
   FOREIGN KEY (FID) REFERENCES FACULTY(FID));
Table created.
```

```
SQL> DESC SUBJECT
Name          Null?    Type
-----        -----
SUB_CODE      NOT NULL VARCHAR2(8)
SNAME          NOT NULL VARCHAR2(20)
TYPE           NOT NULL VARCHAR2(10)
CREDIT          NUMBER(38)
FID            NUMBER(38)
```

### 5. TABLE ENGAGES:

```
SQL> CREATE TABLE ENGAGES
  (FID INT,
   SUB_CODE VARCHAR(10),
   HOURS INT,
   PRIMARY KEY (FID, SUB_CODE),
   FOREIGN KEY (FID) REFERENCES FACULTY(FID),
   FOREIGN KEY (SUB_CODE) REFERENCES SUBJECT(SUB_CODE));
```

Table created.

MI NOTEBOOK

MIAL CAMERA

SQL> DESC ENGAGES

Name	Null?	Type
FID	NOT NULL	NUMBER(38)
SUB_CODE	NOT NULL	VARCHAR2(10)
HOURS		NUMBER(38)

### 6. TABLE TEST:

```
SQL> CREATE TABLE TEST
  (TEST_NO INT,
   TA_DATE DATE,
   TIME INT,
   PRIMARY KEY (TEST_NO));
```

Table created.

```
SQL> DESC TEST
Name Null? Type
-----
TEST_NO NOT NULL NUMBER(38)
IA_DATE DATE
TIME NUMBER(38)
```

#### 7. TABLE SCORE:

```
SQL> CREATE TABLE SCORE
  CUSN VARCHAR(10),
  SUB_CODE VARCHAR(10),
  TEST_NO INT,
  IA_MARKS INT,
  PRIMARY KEY (CUSN, SUB_CODE, TEST_NO),
  FOREIGN KEY (CUSN) REFERENCES STUDENT (CUSN),
  FOREIGN KEY (SUB_CODE) REFERENCES SUBJECT (SUB_CODE),
  FOREIGN KEY (TEST_NO) REFERENCES TEST (TEST_NO);
```

Table created.

```
SQL> DESC SCORE
Name Null? Type
-----
USN NOT NULL VARCHAR2(10)
SUB_CODE NOT NULL VARCHAR2(10)
TEST_NO NOT NULL NUMBER(38)
IA_MARKS NUMBER(38)
```

### 8. TABLE ATTENDANCE :

```
SQL> CREATE TABLE ATTENDANCE
  C USN VARCHAR (10),
  SUB_CODE VARCHAR (8),
  AT_DATE DATE,
  AT_TIME INT,
  STATUS CHAR (1),
  PRIMARY KEY (CUSN, SUB_CODE, AT_DATE),
  FOREIGN KEY (CUSN) REFERENCES STUDENT (CUSN),
  FOREIGN KEY (SUB_CODE) REFERENCES SUBJECT (SUB_CODE);
```

Table created.

Name	Null?	Type
USN	NOT NULL	VARCHAR2(10)
SUB_CODE	NOT NULL	VARCHAR2(8)
AT_DATE	NOT NULL	DATE
AT_TIME		NUMBER(38)
STATUS		CHAR(1)

### 9. TABLE PARENT :

```
SQL> CREATE TABLE PARENT
  C USN VARCHAR (10),
  PNAME VARCHAR (15),
  ADDRESS VARCHAR (15),
  CITY VARCHAR (15),
  PINCODE INT,
  STATE VARCHAR (15),
  PRIMARY KEY (CUSN, PNAME);
```

Table created.

```
SQL> DESC PARENT
Name          Null?    Type
-----          -----
USN           NOT NULL VARCHAR2(10)
PNAME         NOT NULL VARCHAR2(15)
ADDRESS        VARCHAR2(15)
CITY          VARCHAR2(15)
PINCODE       NUMBER(38)
STATE         VARCHAR2(15)
```

Populate each table with appropriate data:

Insert statements for each table

SQL> INSERT INTO FACULTY

VALUES (&FID, '&FNAME', '&LNAME', '&DOB', '&DOJ', '&GENDER',  
'&ADDRESS', '&CITY', &SALARY, '&DESIGNATION', &DNO);

Enter value for fid : 1

Enter value for fname: KULDEEP

Enter value for lname: S

Enter value for dob: 15-DEC-1984

Enter value for doj: 6-AUG-2006

Enter value for gender: M

Enter value for address: ANAGOL

Enter value for city: BELAGAVI

Enter value for salary: 80000

Enter value for designation: ASST. PROF

Enter value for dno: NULL

odd 1: INSERT INTO FACULTY.

VALUES (&FID, '&FNAME', '&LNAME', '&DOB', '&DOJ', '&GENDER',  
'&ADDRESS', '&CITY', &SALARY, '&DESIGNATION', &DNO)

new 1: INSERT INTO FACULTY

VALUES (1, 'KULDEEP', 'S', '15-DEC-1984', '6-AUG-2006', 'M',  
'ANAGOL', 'BELAGAVI', 80000, 'ASST. PROF', NULL)

1 row created.

SQL> SET LINESIZE 250  
SQL> SELECT \* FROM FACULTY;

FID	FNAME	LNAME	DOB	DOJ	G ADDRESS	CITY	SALARY	DESIGNATION	DNO
1	KULDEEP	S	15-DEC-84	06-AUG-06	M ANAGOL	BELGAVI	80000	ASCO.PROF	
2	ARUNDHATI	N	20-DEC-86	10-OCT-08	F RAMNAGAR	GOKAK	60000	ASST.PROF	
3	RANJANA	B	05-JUN-85	18-SEP-09	F SHREENAGAR	HUBLI	62000	ASST.PROF	
4	SUSHANT	M	06-NOV-86	12-JAN-10	M AYODHYANAGAR	ATHAVI	65000	ASCO.PROF	
5	PARIMAL	T	08-NOV-75	16-FEB-00	M TILAKWADI	BELAGAVI	70000	ASCO.PROF	
6	AMRUTA	D	24-FEB-94	02-SEP-16	F CHANNAMMA NAGAR	BELAGAVI	50000	ASST.PROF	

6 rows selected.

SQL> INSERT INTO DEPARTMENT

VALUES (&DNO, '&DNAME', '&DLOC', &HOD\_FID);

Enter value for dno : 10

Enter value for dname : CSE

Enter value for dloc: First floor

Enter value for hod-fid : 1

old \$: INSERT INTO DEPARTMENT

VALUES (&DNO, '&DNAME', '&DLOC', &HOD\_FID);

new \$: INSERT INTO DEPARTMENT

VALUES ( 10, 'CSE', 'First floor', 1 )

1 row created

```
SQL> SELECT * FROM DEPARTMENT
2 ;
```

DNO	DNAME	DLOC	HOD_FID
10	CSE	First floor	1
20	ISE	Second floor	3
30	CSE	First floor	1
40	ECE	Third floor	2
50	CSE	First floor	1
60	EEE	Ground floor	5

6 rows selected.

SQL> UPDATE FACULTY SET DNO=10 WHERE FID=1;

1 row updated

SQL> UPDATE FACULTY SET DNO=20 WHERE FID=2;

1 row updated

SQL> UPDATE FACULTY SET DNO=30 WHERE FID=3;

1 row updated

SQL> UPDATE FACULTY SET DNO=40 WHERE FID=4;

1 row updated

SQL> UPDATE FACULTY SET DNO=50 WHERE FID=5;

1 row updated

SQL> UPDATE FACULTY SET DNO=60 WHERE FID=6;

1 row updated.

FID	FNAME	LNAME	DOB	DOJ	G ADDRESS	CITY	SALARY	DESIGNATION	DNO
1	KULDEEP	S	15-DEC-84	06-AUG-06	M ANAGOL	BELGAVI	80000	ASCO.PROF	10
2	ARUNDHATI	N	28-DEC-86	10-OCT-08	F RAMNIGAR	GOKAK	60000	ASST.PROF	20
3	RANJANA	B	05-JUN-85	18-SEP-09	F SHREENAGAR	HUBLI	62000	ASST.PROF	30
4	SUSHANT	M	06-NOV-86	12-JAN-10	M AYODHYANAGAR	ATHANI	65000	ASCO.PROF	40
5	PARIMAL	T	08-NOV-75	16-FEB-00	M TILAKWADI	BELAGAVI	70000	ASCO.PROF	50
6	AMRUTA	D	24-FEB-94	02-SEP-16	F CHANNAMMA NAGAR	BELAGAVI	50000	ASST.PROF	60

6 rows selected.

```
SQL> INSERT INTO STUDENT
VALUES ('&USN', '&FNAME', '&LNAME', '&GENDER', '&ADDRESS',
&SEM, &MOB_NO, '&BDATE', &DNO);
```

Enter value for USN : 2GI18CS001

Enter value for fname: RAVI

Enter value for lname: PATIL

Enter value for gender: M

Enter value for address: Raibag

Enter value for sem: 4

Enter value for mob-no : 9850783092

Enter value for bdate : 14-MAR-2001

Enter value for dno : 10.

old 1 : INSERT INTO STUDENT

```
VALUES ('&USN', '&FNAME', '&LNAME', '&GENDER', '&ADDRESS',
&SEM, &MOB_NO, '&BDATE', &DNO);
```

new 1 : INSERT INTO STUDENT

```
VALUES ('2GI18CS001', 'RAVI', 'PATIL', 'M', 'RAIBAG', 4,
9850783092, '14-MAR-2001', 10).
```

1 row created

```
SQL> SET LINESIZE 250
SQL> SELECT * FROM STUDENT;
```

USN	FNAME	LNAME	G ADDRESS	SEM	MOB_NO	BDATE	DNO
2GI18CS001	RAVI	PATIL	M RAILBAG	4	9850783092	14-MAR-01	10
2GI18CS002	RAJAT	PUJARI	M ATHANI	4	9448222904	03-APR-02	10
2GI18CS003	RAMYA	SHERI	F HUBLI	4	1234567890	01-MAY-02	20
2GI18CS004	ASHA	BHAT	F UDUPI	4	7463457288	23-DEC-02	20
2GI18CS005	RESHMA	UPPAR	F DHARWAD	4	987654321	10-JUL-02	40

SQL> INSERT INTO SUBJECT  
 VALUES ('&SUB\_CODE', '&SNAME', '&TYPE', &CREDIT, &FID);  
 Enter value for sub\_code: CS41  
 Enter value for Sname: DBMS  
 Enter value for type: CORE  
 Enter value for credit: 4  
 Enter value for fid: 1  
 old : INSERT INTO &SUBJECT  
 VALUES ('&SUB\_CODE', '&SNAME', '&TYPE', &CREDIT, &FID)  
 new : INSERT INTO SUBJECT VALUES ('CS41', 'DBMS',  
 'CORE', 4, 1)

, now created.

SQL> SELECT \* FROM SUBJECT;

SUB_CODE	SNAME	TYPE	CREDIT	FID
CS41	DBMS	CORE	4	1
CS42	OS	CORE	3	2
CS43	PYTHON	INTEGRATED	2	3
CS44	SE	ELECTIVE	3	4
CS45	MATHS	CORE	4	5

SQL> INSERT INTO TEST VALUES ('TEST\_NO', 'IA\_DATE',  
 & TIME);

Enter value for test\_no : 1

Enter value for ia\_date: 31-MAY-2021

Enter value for time: 10

old 1: INSERT INTO TEST VALUES ('TEST\_NO', 'IA\_DATE',  
 & TIME);

new 1: INSERT INTO TEST VALUES (1, '31-MAY-2021', 10)

, now created .

SQL> SELECT \* FROM TEST;

TEST_NO	IA_DATE	TIME
1	31-MAY-21	10
2	01-JUN-21	10
3	01-JUN-21	2
4	02-JUN-21	10
5	02-JUN-21	2

SQL> INSERT INTO SCORE VALUES  
 ('&USN', '&SUB-CODE', &TEST\_NO, &IA\_MARKS);  
 Enter value for usn: 2GI18CS001  
 Enter value for sub-code: CS41  
 Enter value for test-no: 1  
 Enter value for ia-marks: 25.  
  
 old \$: INSERT INTO SCORE VALUES  
 ('&USN', '&SUB-CODE', &TEST\_NO, &IA\_MARKS);  
 new \$: INSERT INTO SCORE VALUES  
 ('2GI18CS001', 'CS41', 1, 25)

USN	SUB_CODE	TEST_NO	IA_MARKS
2GI18CS001	CS41	1	25
2GI18CS002	CS42	1	24
2GI18CS003	CS43	2	25
2GI18CS004	CS44	2	23
2GI18CS005	CS45	3	25

SQL> INSERT INTO ATTENDANCE  
 VALUES ('&USN', '&SUB-CODE', '&AT-DATE', '&AT-TIME', '&STATUS');  
 Enter value for usn: 2GI18CS001  
 Enter value for sub-code: CS41  
 Enter value for at-date: 22-FEB-2021  
 Enter value for at-time: 10  
 Enter value for status: P.  
  
 old 1 : INSERT INTO ATTENDANCE VALUES  
 ( '&USN', '&SUB-CODE', '&AT-DATE', '&AT-TIME', '&STATUS');  
 new 1 : INSERT INTO ATTENDANCE VALUES  
 ( '2GI18CS001', 'CS41', '22-FEB-2021', 10, 'P')  
 1 row created

SQL> SELECT \* FROM ATTENDANCE;

USN	SUB_CODE	AT_DATE	AT_TIME	S
2GI18CS001	CS41	22-FEB-21	10	P
2GI18CS002	CS42	23-FEB-21	2	P
2GI18CS003	CS43	24-FEB-21	10	P
2GI18CS004	CS44	25-FEB-21	2	A
2GI18CS005	CS45	26-FEB-21	10	A

```
SQLY INSERT INTO PARENT
VALUES ('+USN','+PNAME','+ADDRESS','+CITY'+PINCODE,
       '+STATE');
```

Enter value for usn: 2GI18CS001.

Enter value for pname: SURESH

Enter value for address: FIRST CROSS

Enter value for city: BELAGAVI

Enter value for pincode: 546892

Enter value for state: KARNATAKA

old 1 : INSERT INTO PARENT

```
VALUES ('+USN','+PNAME','+ADDRESS','+CITY',+PINCODE,
       '+STATE')
```

new 1 : INSERT INTO PARENT

```
VALUES ('2GI18CS001','SURESH','FIRST CROSS','BELAGAVI',
       '546892','KARNATAKA')
```

SQL> SELECT \* FROM PARENT;

USN	PNAME	ADDRESS	CITY	PINCODE	STATE
2GI18CS001	SURESH	FIRSTCROSS	BELAGAVI	546892	KARNATAKA
2GI18CS002	PRAMILA	SECONDCROSS	DHARWAD	637854	KARNATAKA
2GI18CS003	PRATHAMESH	RAMNAGAR	HUBLI	645389	KARNATAKA
2GI18CS004	SUNITA	SHANTINAGAR	ATHANI	834589	KARNATAKA
2GI18CS005	SANDHYA	SHREENAGAR	GOKAK	756389	KARNATAKA

### QUERIES:

1. Compute the total number of male and female students in each semester.

```

1) SELECT GENDER, COUNT GENDER AS COUNT
2) FROM STUDENT
3) GROUP BY GENDER.

```

G	COUNT
M	2
F	3

2. Display the details of Faculty who were joined during 2000's.

```

1) SELECT *
2) FROM FACULTY
3) WHERE DOJ LIKE '____-1-';

```

FID	FNAME	LNAME	DOB	DOJ	G ADDRESS	CITY	SALARY	DISGNATION
<hr/>								
40	D SUSHANT	M	06-NOV-86	12-JAN-10	M AYODHYANAGAR	ATHANI	65000	ASCO.PROF
40	AMRUTA	D	24-FEB-94	02-SEP-16	F CHENNAMANAGAR	BELAGAVI	50000	ASST.PROF

3. Show the effect of giving of 20% raise in salary where faculty ID is 2.

```

1) SELECT SALARY * 1.2
2) FROM FACULTY
3) WHERE FID = 2 ;

```

SALARY\*1.2

72000

4. Retrieve all Parent details whose address pin code ends with 9.

```
SQL> SELECT *
2   FROM PARENT
3 WHERE PINCODE LIKE '%.1.9';
```

USN	PNAME	ADDRESS	CITY	PINCODE	STATE
2GI18CS003	PRATHMESH	RAMNAGAR	HUBLI	645389	KARNATAKA
2GI18CS004	SUNITA	SHANTINAGAR	ATHANI	834589	KARNATAKA
2GI18CS005	SANDHYA	SHREENAGAR	GOKAK	756389	KARNATAKA

### Graphical User Interface:

### Outputs:

FID	FNA...	LNA...	DOB	DOJ	GEN...	ADD...	CITY	SALA...	DESI...	DNO
1	KUL...	S	1984...	2006...	M	ANA...	BEL...	80000	ASC...	10
2	ARU...	N	1986...	2008...	F	RAM...	GOK...	60000	ASS...	20
3	RAN...	B	1985...	2009...	F	SHR...	HUBLI	62000	ASS...	30
4	SUS...	M	1986...	2010...	M	AYO...	ATH...	65000	ASC...	40
5	PARI...	T	1975...	2000...	M	TILA...	BEL...	70000	ASC...	50
6	AMR...	D	1994...	2016...	F	CHA...	BEL...	50000	ASS...	60

Educational Institute

Select the Table to be displayed

DEPARTMENT	▼
<b>Submit</b>	

DEPARTMENT TABLE

DNO	DNAME	DLOC	HOD_FID
10	CSE	First floor	1
20	ISE	Second floor	3
30	CSE	First floor	1
40	ECE	Third floor	2
50	CSE	First floor	1
60	EEE	Ground floor	5

**OK**

Educational Institute

Select the Table to be displayed

STUDENT	▼
<b>Submit</b>	

STUDENT TABLE

USN	FNAME	LNAME	GEND...	ADDR...	SEM	MOB...	BDATE	DNO
2GI18...	RAVI	PATIL	M	RAILB...	4	98507...	2001-...	10
2GI18...	RAJAT	PUJARI	M	ATHANI	4	94482...	2002-...	10
2GI18...	RAMYA	SHERI	F	HUBLI	4	12345...	2002-...	20
2GI18...	ASHA	BHAT	F	UDUPI	4	74634...	2002-...	20
2GI18...	RESH...	UPPAR	F	DHAR...	4	98765...	2002-...	40

**OK**

Educational Institute

Select the Table to be displayed

SUBJECT

Submit

SUBJECT TABLE

SUB_CODE	SNAME	TYPE	CREDIT	FID
CS41	DBMS	CORE	4	1
CS42	OS	CORE	3	2
CS43	PYTHON	INTEGRATED	2	3
CS44	SE	ELECTIVE	3	4
CS45	MATHS	CORE	4	5

OK

Educational Institute

Select the Table to be displayed

TEST

Submit

TEST TABLE

TEST_NO	IA_DATE	TIME
1	2021-05-31 00:00:00.0	10
2	2021-06-01 00:00:00.0	10
3	2021-06-01 00:00:00.0	2
4	2021-06-02 00:00:00.0	10
5	2021-06-02 00:00:00.0	2

OK



Select the Table to be displayed

SCORE

Submit

SCORE TABLE

USN	SUB_CODE	TEST_NO	IA_MARKS
2GI18CS001	CS41	1	25
2GI18CS002	CS42	1	24
2GI18CS003	CS43	2	25
2GI18CS004	CS44	2	23
2GI18CS005	CS45	3	25

OK



Select the Table to be displayed

ATTENDANCE

Submit

ATTENDANCE TABLE

USN	SUB_CODE	AT_DATE	AT_TIME	STATUS
2GI18CS001	CS41	2021-02-22 0...	10	P
2GI18CS002	CS42	2021-02-23 0...	2	P
2GI18CS003	CS43	2021-02-24 0...	10	P
2GI18CS004	CS44	2021-02-25 0...	2	A
2GI18CS005	CS45	2021-02-26 0...	10	A

OK

## Conclusion:

We learnt to use the open-source ER-design tool dia and created the ER-Model for the above said problem statement. Converted the ER-diagram into relational schema diagram by applying the ER- to relational rules. We identified primary and foreign keys and created all the relations in Oracle DBMS using DDL statements. Further, the database was populated with real data using insert statements. The content of each table was displayed using SELECT sql statement. We learnt about update and delete sql statements and also learnt about alter table command to modify or add constraints to the table structure after they are created. Learnt how a GUI can be built to connect to a back-end database in ORACLE / MYSQL using JAVA / PHP.

## **Design of Airline Flight Information Database**

Expt no: 3

Date: 09-06-2021

### **Problem Statement:**

Consider the schema for airline flight information Database:

**FLIGHTS (no: integer, fromPlace: string, toPlace: string, distance: integer, Departs: date, arrives: date, price: real)**

**AIRCRAFT (aid: integer, aname: string, cruisingrange: integer)**

**CERTIFIED (eid: integer, aid: integer)**

**EMPLOYEES (eid: integer, ename: string, salary: integer)**

Create tables and populate with appropriate values (At least 5 records in each table) for the given database.

Write SQL queries to:

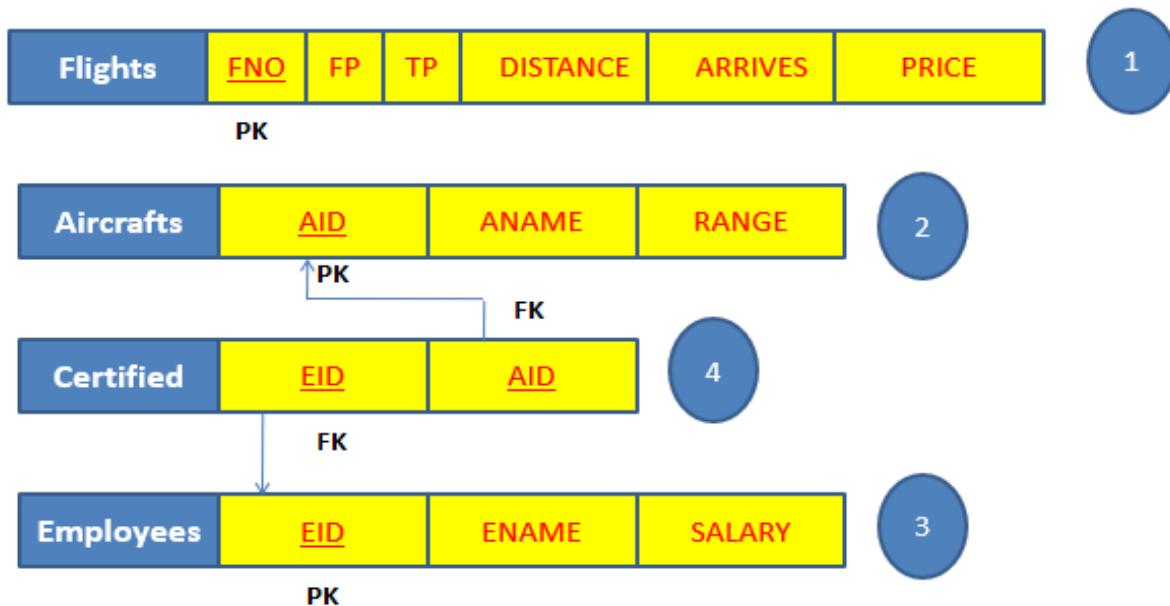
1. Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs. 80000/-.
2. For each pilot who is certified for more than three aircrafts, find the eid, ename and the maximum cruising range of the aircraft for which she or he is certified.
3. Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Frankfurt
4. Find the aids of all aircrafts that can be used on routes from Bengaluru to New Delhi.

Concepts: ER-Modelling is a first step towards building a database application. It helps in identifying various entities, their attributes and the relationship between them. The ER-Model helps the application developers to explain the customers, what all data would be stored and seek their suggestions to include all the data relevant to the application. While designing an ER-Model it is important to include only the attributes relevant to the entity types. Further after drawing ER-diagram the structural constraints, namely the cardinality ratios and participation constraints must be correctly indicated. There 7 mapping rules, which must be applied to the ER-diagram after completion to get the Relational model. The relational schema diagram for each relation must be drawn and the Primary key and foreign keys must be correctly indicated. Then the DDL statements must be used to create the tables in Oracle DBMS. Using INSERT command data must be inserted and using SQL queries the data must be checked for its correctness.

### Learning Outcomes:

- 17.ER-model is a set of concepts to describe data in graphical form.
- 18.There are 7 ER-to-Relational mapping rules to get Relational model from ER-Model.
19. Relational model is a set of concepts to describe data to RDBMS.
- 20.Relation, tuple, attribute, domain, Primary key, foreign key are the concepts in Relational mode.
- 21.DDL statements help us to create tables and specify constraints.
- 22.DML statements help us to populate and manipulate the database.
- 23.Design a GUI and write the program to connect to DBMS Server and display the data.
- 24 Learned to use SQL queries to list data stored in tables.

## Schema Diagram:



## Create Table Statements:

### 1. TABLE FLIGHTS:

```
SQL> CREATE TABLE FLIGHTS
(FNO INTEGER,
FROMPLACE VARCHAR (20) NOT NULL,
TOPLACE VARCHAR (20) NOT NULL,
DISTANCE INTEGER,
DEPARTS DATE NOT NULL,
ARRIVES DATE NOT NULL,
PRICE REAL,
PRIMARY KEY(FNO));
```

Name	Null?	Type
FNO	NOT NULL	NUMBER(38)
FROMPLACE		VARCHAR2(20)
TOPLACE		VARCHAR2(20)
DISTANCE		NUMBER(38)
DEPARTS		DATE
ARRIVES		DATE
PRICE		FLOAT(63)

## 2. TABLE AIRCRAFT:

```
SQL> CREATE TABLE AIRCRAFT
  (AID INTEGER,
   ANAME VARCHAR (15),
   CRUISINGRANGE INTEGER,
   PRIMARY KEY(AID));
```

Table created.

Name	Null?	Type
AID	NOT NULL	NUMBER(38)
ANAME		VARCHAR2(15)
CRUISINGRANGE		NUMBER(38)

## 3. TABLE EMPLOYEES:

```
SQL> CREATE TABLE EMPLOYEES
  (EID INTEGER,
   ENAME VARCHAR (15),
   SALARY INTEGER,
   PRIMARY KEY(EID));
```

Table created.

Name	Null?	Type
EID	NOT NULL	NUMBER(38)
ENAME		VARCHAR2(15)
SALARY		NUMBER(38)

#### 4. TABLE CERTIFIED:

```
SQL> CREATE TABLE CERTIFIED
      (EID INTEGER,
       AID INTEGER,
       PRIMARY KEY (EID, AID),
       FOREIGN KEY (EID) REFERENCES EMPLOYEES(EID),
       FOREIGN KEY(AID) REFERENCES AIRCRAFT(AID));
```

Table created.

Name	Null?	Type
EID	NOT NULL	NUMBER(38)
AID	NOT NULL	NUMBER(38)

#### INSERT Statements for each Table:

1. INSERT INTO FLIGHTS VALUES  
(255,'BANGALORE','FRANKFURT',500,'08-AUG-2011','10-AUG-2011',8000)
2. INSERT INTO FLIGHTS VALUES  
(256,'BANGALORE','DELHI',200,'03-SEP-2016','04-SEP-2016',4000)
3. : INSERT INTO FLIGHTS VALUES  
(257,'BANGALORE','MANGALORE',200,'12-JUN-2018','12-JUN-2018',1000)
4. : INSERT INTO FLIGHTS VALUES  
(258,'BANGALORE','MUMBAI',400,'15-DEC-2012','15-DEC-2012',2000)

5. INSERT INTO FLIGHTS VALUES  
(259,'BANGALORE','FRANKFURT',500,'09-FEB-2003','11-FEB-2003',8000)
6. : INSERT INTO FLIGHTS VALUES  
(260,'BANGALORE','DELHI',300,'05-OCT-2007','06-OCT-2007',5000)

```
SQL> SET LINESIZE 150
SQL> SELECT * FROM FLIGHTS;

      FNO FROMPLACE          TOPLACE      DISTANCE DEPARTS ARRIVES    PRICE
-----  -----  -----  -----  -----  -----  -----
      255 BANGALORE        FRANKFURT      500 08-AUG-11 10-AUG-11   8000
      256 BANGALORE        DELHI         200 03-SEP-16 04-SEP-16   4000
      257 BANGALORE        MANGALORE     200 12-JUN-18 12-JUN-18   1000
      258 BANGALORE        MUMBAI        400 15-DEC-12 15-DEC-12   2000
      259 BANGALORE        FRANKFURT     500 09-FEB-03 11-FEB-03   8000
      260 BANGALORE        DELHI         300 05-OCT-07 06-OCT-07   5000

6 rows selected.
```

1. INSERT INTO AIRCRAFT VALUES (685,'BOEING15',1000)
2. INSERT INTO AIRCRAFT VALUES (686,'BOEING10',2000)
3. INSERT INTO AIRCRAFT VALUES (687,'SKYTRAIN',1000)
4. INSERT INTO AIRCRAFT VALUES (688,'AVENGER',900)
5. INSERT INTO AIRCRAFT VALUES (689,'BOEING12',2000)

```
SQL> SELECT * FROM AIRCRAFT;

      AID ANAME          CRUISINGRANGE
-----  -----  -----
      685 BOEING15        1000
      686 BOEING10        2000
      687 SKYTRAIN        1000
      688 AVENGER         900
      689 BOEING12        2000
```

1. INSERT INTO EMPLOYEES VALUES (101,'RAMESH',90000)
2. INSERT INTO EMPLOYEES VALUES (102,'AMIT',85000)
3. INSERT INTO EMPLOYEES VALUES (103,'PRADEEP',70000)
4. INSERT INTO EMPLOYEES VALUES (104,'RAMYA',60000)

5. INSERT INTO EMPLOYEES VALUES (105,'SAMRUDHI',75000)

SQL> SELECT * FROM EMPLOYEES;		
EID	ENAME	SALARY
101	RAMESH	90000
102	AMIT	85000
103	PRADEEP	70000
104	RAMYA	60000
105	SAMRUDHI	75000

1. INSERT INTO CERTIFIED VALUES (101,686)
2. INSERT INTO CERTIFIED VALUES (101,687)
3. INSERT INTO CERTIFIED VALUES (102,685)
4. INSERT INTO CERTIFIED VALUES (103,688)
5. INSERT INTO CERTIFIED VALUES (104,689)
6. INSERT INTO CERTIFIED VALUES (105,688)

SQL> SELECT * FROM CERTIFIED;		
EID	AID	
101	685	
101	686	
101	687	
102	685	
103	688	
104	689	
105	688	

7 rows selected.

## SQL Queries:

- Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80,000.

```
SQL> SELECT DISTINCT A.aname
  2  FROM AIRCRAFT A, CERTIFIED C, EMPLOYEES E
  3  WHERE A.AID = C.AID AND C.EID = E.EID AND E.SALARY >80000;

ANAME
-----
BOEING10
SKYTRAIN
BOEING15
```

- For each pilot who is certified for more than three aircrafts, find the eid, ename and the maximum cruising range of the aircraft for which she or he is certified.

```
SQL> SELECT      C.EID,    MAX(A.CRUISINGRANGE)
  2  FROM      CERTIFIED  C,  AIRCRAFT A
  3  WHERE      C.AID = A.AID
  4  GROUP BY   C.EID
  5  HAVING     COUNT(*) > = 2;

          EID  MAX(A.CRUISINGRANGE)
-----  -----
        101           2000
```

- Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Frankfurt.

```
SQL> SELECT E.ENAME
  2  FROM EMPLOYEES E
  3  WHERE E.SALARY <  (SELECT MIN(PRICE)
  4  FROM FLIGHTS
  5  WHERE FROMPLACE = 'BANGALORE' AND TOPLACE = 'FRANKFURT');

no rows selected
```

4. Find the names of pilots whose salary is greater than the price of the cheapest route from Bengaluru to Frankfurt.

```
SQL> SELECT E.ENAME
  2  FROM EMPLOYEES E
  3 WHERE E.SALARY > (SELECT MIN(PRICE)
  4                   FROM FLIGHTS
  5                   WHERE FROMPLACE = 'BANGALORE' AND TOPLACE = 'FRANKFURT');

ENAME
-----
RAMESH
AMIT
PRADEEP
RAMYA
SAMRUDHI
```

5. Find the aids of all aircraft that can be used on routes from Bengaluru to New Delhi.

```
SQL> SELECT A.AID
  2  FROM AIRCRAFT A
  3 WHERE A.CRUISINGRANGE > (SELECT MIN(DISTANCE)
  4                               FROM FLIGHTS
  5                               WHERE FROMPLACE='BANGALORE' AND TOPLACE = 'DELHI');

AID
-----
685
686
687
688
689
```

6. Find the names of pilots certified for some Boeing aircraft.

```
SQL> SELECT DISTINCT E.ENAME
  2  FROM AIRCRAFT A, CERTIFIED C, EMPLOYEES E
  3 WHERE A.AID = C.AID AND C.EID = E.EID AND A.ANAME LIKE 'BOEING%';

ENAME
-----
RAMYA
AMIT
RAMESH
```

## Conclusion:

We learnt to use the open-source ER-design tool dia and created the ER-Model for the above said problem statement. Converted the ER-diagram into relational schema diagram by applying the ER- to relational rules. We identified primary and foreign keys and created all the relations in Oracle DBMS using DDL statements. Further, the database was populated with real data using insert statements. The content of each table was displayed using SELECT sql statement. We learnt about update and delete sql statements and also learnt about alter table command to modify or add constraints to the table structure after they are created.

## **Design of ER-Model for an Order Database**

Expt no: 4

Date: 17-06-2021

### **Problem Statement:**

Consider the following schema for Order Database:

SALESMAN (*Salesman\_id, Name, City, Commission*)

CUSTOMER (*Customer\_id, Cust\_Name, City, Grade, Salesman\_id*)

ORDERS (*Ord\_No, Purchase\_Amt, Ord\_Date, Customer\_id, Salesman\_id*)

Create tables and populate with appropriate values (Atleast 5 records in each table) for the given database

Write SQL queries to

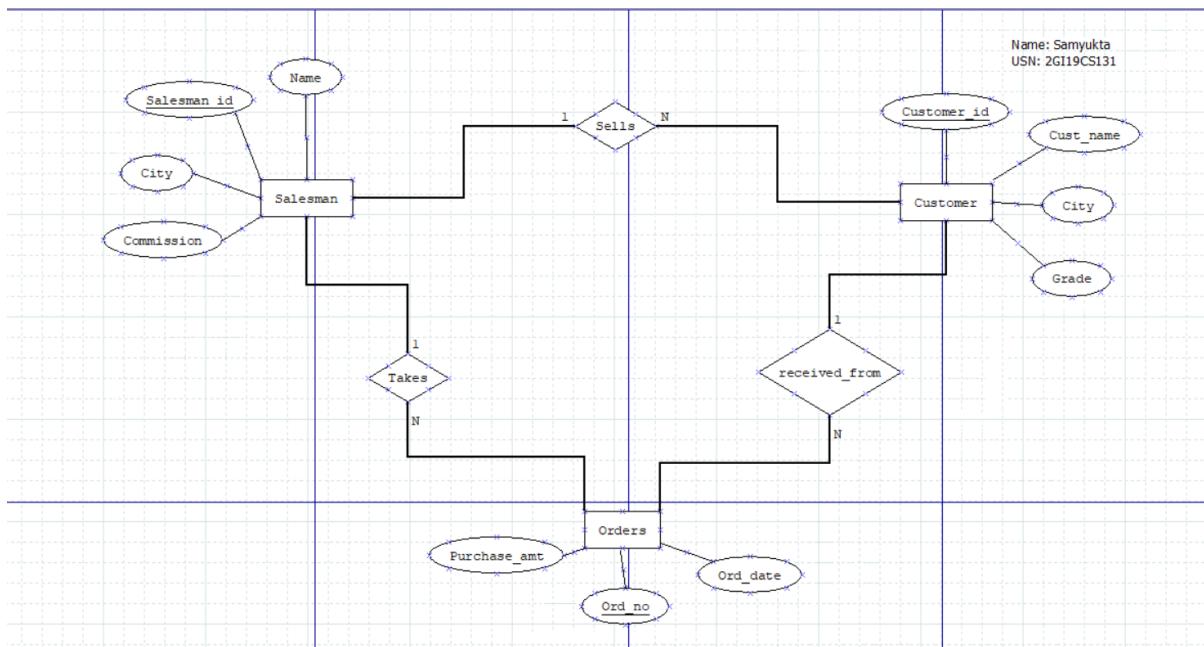
1. Count the customers with grades above Bangalore's average.
2. Find the name and numbers of all salesmen who had more than one customer.
3. List all the salesmen names and customer names whom order amount is more than 4000.
4. Demonstrate the DELETE operation by removing salesman with id 1000.  
All his orders must also be deleted.

Concepts: ER-Modelling is a first step towards building a database application. It helps in identifying various entities, their attributes and the relationship between them. The ER-Model helps the application developers to explain the customers, what all data would be stored and seek their suggestions to include all the data relevant to the application. While designing an ER-Model it is important to include only the attributes relevant to the entity types. Further after drawing ER-diagram the structural constraints, namely the cardinality ratios and participation constraints must be correctly indicated. There 7 mapping rules, which must be applied to the ER-diagram after completion to get the Relational model. The relational schema diagram for each relation must be drawn and the Primary key and foreign keys must be correctly indicated. Then the DDL statements must be used to create the tables in Oracle DBMS. Using INSERT command data must be inserted and using SQL queries the data must be checked for its correctness.

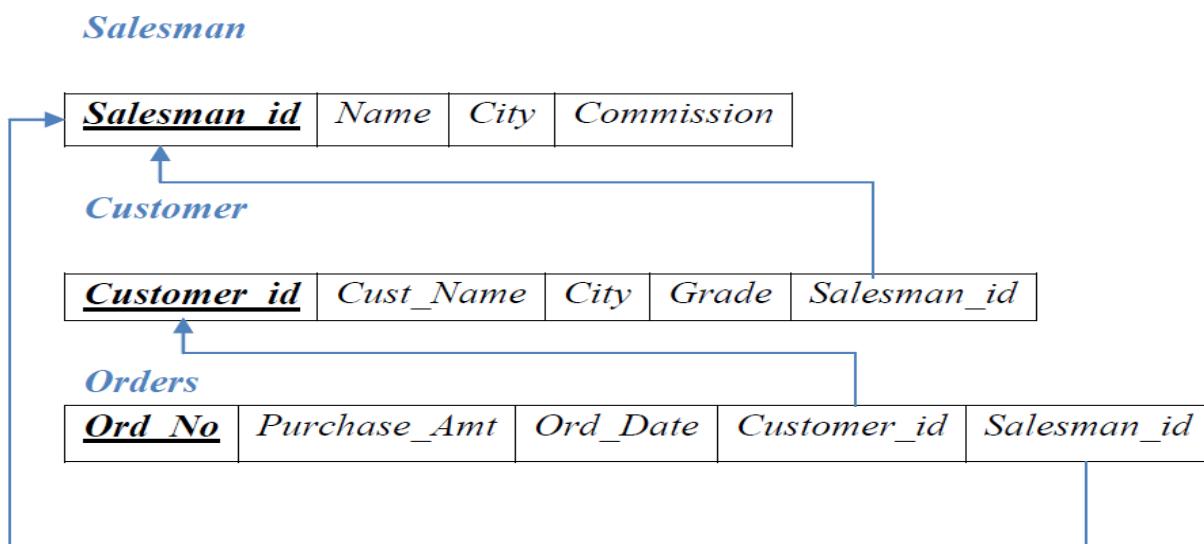
### Learning Outcomes:

- 25.ER-model is a set of concepts to describe data in graphical form.
- 26.There are 7 ER-to-Relational mapping rules to get Relational model from ER-Model.
27. Relational model is a set of concepts to describe data to RDBMS.
- 28.Relation, tuple, attribute, domain, Primary key, foreign key are the concepts in Relational mode.
- 29.DDL statements help us to create tables and specify constraints.
- 30.DML statements help us to populate and manipulate the database.
- 31.Design a GUI and write the program to connect to DBMS Server and display the data.
- 32.Learned to use SQL queries to list data stored in tables.

### ER Diagram:



### Schema Diagram:



## Create Table Statements:

```
SQL> CREATE TABLE SALESMAN (
    SALESMAN_ID INT,
    NAME VARCHAR(10) NOT NULL,
    CITY VARCHAR(10),
    COMMISSION INT NOT NULL,
    PRIMARY KEY(SALESMAN_ID)
);
```

Table created.

Name	Null?	Type
SALESMAN_ID	NOT NULL	NUMBER(38)
NAME	NOT NULL	VARCHAR2(10)
CITY		VARCHAR2(10)
COMMISSION	NOT NULL	NUMBER(38)

```
SQL> CREATE TABLE CUSTOMER (
    CUSTOMER_ID INT,
    CUST_NAME VARCHAR(10) NOT NULL,
    CITY VARCHAR(10),
    GRADE INT,
    SALESMAN_ID INT,
    PRIMARY KEY (CUSTOMER_ID),
    FOREIGN KEY (SALESMAN_ID) REFERENCES SALESMAN
    (SALESMAN_ID) ON DELETE SET NULL);
```

Table created

```
SQL> DESC CUSTOMER
```

Name	Null?	Type
CUSTOMER_ID	NOT NULL	NUMBER(38)
CUST_NAME	NOT NULL	VARCHAR2(10)
CITY		VARCHAR2(10)
GRADE		NUMBER(38)
SALESMAN_ID		NUMBER(38)

```
SQL> CREATE TABLE ORDERS (
```

```
    ORD_NO INT,
    PURCHASE_AMT INT,
    ORD_DATE DATE NOT NULL,
    CUSTOMER_ID INT,
    SALESMAN_ID INT,
    PRIMARY KEY (ORD_NO),
    FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMER
    (CUSTOMER_ID) ON DELETE CASCADE,
    FOREIGN KEY (SALESMAN_ID) REFERENCES SALESMAN
    (SALESMAN_ID) ON DELETE CASCADE
);
```

Table created.

```
SQL> DESC ORDERS
```

Name	Null?	Type
ORD_NO	NOT NULL	NUMBER(38)
PURCHASE_AMT		NUMBER(38)
ORD_DATE	NOT NULL	DATE
CUSTOMER_ID		NUMBER(38)
SALESMAN_ID		NUMBER(38)

INSERT Statements for each Table:

```
INSERT INTO SALESMAN VALUES (1000, 'JOHN','BANGLORE','25');
```

1 row created.

```
SQL> INSERT INTO SALESMAN VALUES (1001,  
'RAVI','BANGALORE','20');
```

1 row created.

```
SQL> INSERT INTO SALESMAN VALUES (1002,  
'KUMAR','MYSORE','30');
```

1 row created.

```
SQL> INSERT INTO SALESMAN VALUES (1003, 'RAJ','HUBLI','35');
```

1 row created.

```
SQL> INSERT INTO SALESMAN VALUES (1004,  
'VICKY','BELGUM','40');
```

1 row created.

SQL> SELECT * FROM SALESMAN;			
SALESMAN_ID	NAME	CITY	COMMISSION
1000	JOHN	BANGLORE	25
1001	RAVI	BANGALORE	20
1002	KUMAR	mysore	30
1003	RAJ	HUBLI	35
1004	VICKY	BELGUM	40

INSERT INTO CUSTOMER VALUES (1, 'SACHI','BANGLORE',100,1000);

1 row created.

SQL> INSERT INTO CUSTOMER VALUES (2, 'MEEN','MANGALORE',300,1000);

1 row created.

SQL> INSERT INTO CUSTOMER VALUES (3, 'VEENA','HUBLI',400,1001);

1 row created.

SQL> INSERT INTO CUSTOMER VALUES (4, 'LEENA','BANGLORE',500,1002);

1 row created.

```
SQL>      INSERT      INTO      CUSTOMER      VALUES      (5,  
'PRIYA','BELGUM',400,1001);
```

1 row created.

SQL> SELECT * FROM CUSTOMER;				
CUSTOMER_ID	CUST_NAME	CITY	GRADE	SALESMAN_ID
1	SACHI	BANGLORE	100	1000
2	MEEN	MANGALORE	300	1000
3	VEENA	HUBLI	400	1001
4	LEENA	BANGLORE	500	1002
5	PRIYA	BELGUM	400	1001

```
INSERT INTO ORDERS VALUES (111, 500, '01-JAN-21', 2, 1001);
```

1 row created.

```
SQL> INSERT INTO ORDERS VALUES (222, 600, '02-JAN-21', 2, 1001);
```

1 row created.

```
SQL> INSERT INTO ORDERS VALUES (333, 4000, '05-FEB-21', 1, 1001);
```

1 row created.

```
SQL> INSERT INTO ORDERS VALUES (444, 5000, '06-MAR-21', 1, 1000);
```

1 row created.

SQL> INSERT INTO ORDERS VALUES (555, 6000, '20-DEC-20', 1, 1000);

1 row created.

ORD_NO	PURCHASE_AMT	ORD_DATE	CUSTOMER_ID	SALESMAN_ID
111	500	01-JAN-21	2	1001
222	600	02-JAN-21	2	1001
333	4000	05-FEB-21	1	1001
444	5000	06-MAR-21	1	1000
555	6000	20-DEC-20	1	1000

## SQL Queries:

1. Count the customers with grades above Bangalore's average.

SQL> SELECT AVG(GRADE)  
FROM CUSTOMER  
WHERE CITY = 'BANGLORE';

AVG(GRADE)
300

```
SQL> SELECT COUNT(CUSTOMER_ID)
  FROM CUSTOMER
 WHERE GRADE > (SELECT AVG(GRADE)
  FROM CUSTOMER
 WHERE CITY = 'BANGLORE');
```

```
SQL> SELECT COUNT(CUSTOMER_ID)
 2  FROM CUSTOMER
 3  WHERE GRADE > (SELECT AVG(GRADE)
 4                      FROM CUSTOMER
 5                      WHERE CITY = 'BANGLORE');

COUNT(CUSTOMER_ID)
-----
3
```

- Find the name and numbers of all salesmen who had more than one customer.

```
SQL> SELECT S.SALESMAN_ID, S.NAME
  FROM SALESMAN S
 WHERE (SELECT COUNT (*) FROM CUSTOMER C
 WHERE C.SALESMAN_ID=S.SALESMAN_ID) > 1;
```

```
SQL> SELECT S.SALESMAN_ID, S.NAME
 2  FROM SALESMAN S
 3  WHERE (SELECT COUNT (*)
 4          FROM CUSTOMER C
 5          WHERE C.SALESMAN_ID=S.SALESMAN_ID) > 1;

SALESMAN_ID  NAME
-----  -----
    1000  JOHN
    1001  RAVI
```

USING GROUP BY AND HAVING CLAUSE:

```
SQL> SELECT S.NAME,COUNT(CUSTOMER_ID)
  2  FROM SALESMAN S,CUSTOMER C
  3  WHERE S.SALESMAN_ID = C.SALESMAN_ID
  4  GROUP BY S.NAME
  5  HAVING COUNT(CUSTOMER_ID)>1;

NAME          COUNT(CUSTOMER_ID)
-----        -----
RAVI           2
JOHN           2
```

3. List all the salesmen names and customer names whom order amount is more than 4000.

```
SQL> SELECT S.NAME,C.CUST_NAME
      FROM CUSTOMER C,SALESMAN S,ORDERS O
      WHERE S.SALESMAN_ID=O.SALESMAN_ID AND
      C.CUSTOMER_ID=O.CUSTOMER_ID AND
      O.PURCHASE_AMT > 4000;
```

NAME	CUST_NAME
JOHN	SACHI
JOHN	SACHI

4. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.

```
SQL> DELETE FROM SALESMAN WHERE SALESMAN_ID=1000;
```

```
SQL> select * from salesman;
```

SALESMAN_ID	NAME	CITY	COMMISSION
1001	ravi	banglore	20
1002	kumar	mysore	30
1003	raj	hubli	35
1004	vicky	belgum	40

### Conclusion:

We learnt to use the open-source ER-design tool dia and created the ER-Model for the above said problem statement. Converted the ER-diagram into relational schema diagram by applying the ER- to relational rules. We identified primary and foreign keys and created all the relations in Oracle DBMS using DDL statements. Further, the database was populated with real data using insert statements. The content of each table was displayed using SELECT sql statement. We learnt about update and delete sql statements and also learnt about alter table command to modify or add constraints to the table structure after they are created.

## **Design of ER-Model for an Movie Database**

Expt no: 5

Date: 23-06-2021

### **Problem Statement:**

Consider the schema for Movie Database:

ACTOR (*Act\_id, Act\_Name, Act\_Gender*)

DIRECTOR (*Dir\_id, Dir\_Name, Dir\_Phone*)

MOVIES (*Mov\_id, Mov\_Title, Mov\_Year, Mov\_Lang, Dir\_id*)

MOVIE\_CAST (*Act\_id, Mov\_id, Role*)

RATING (*Mov\_id, Rev\_Stars*)

Write SQL queries to

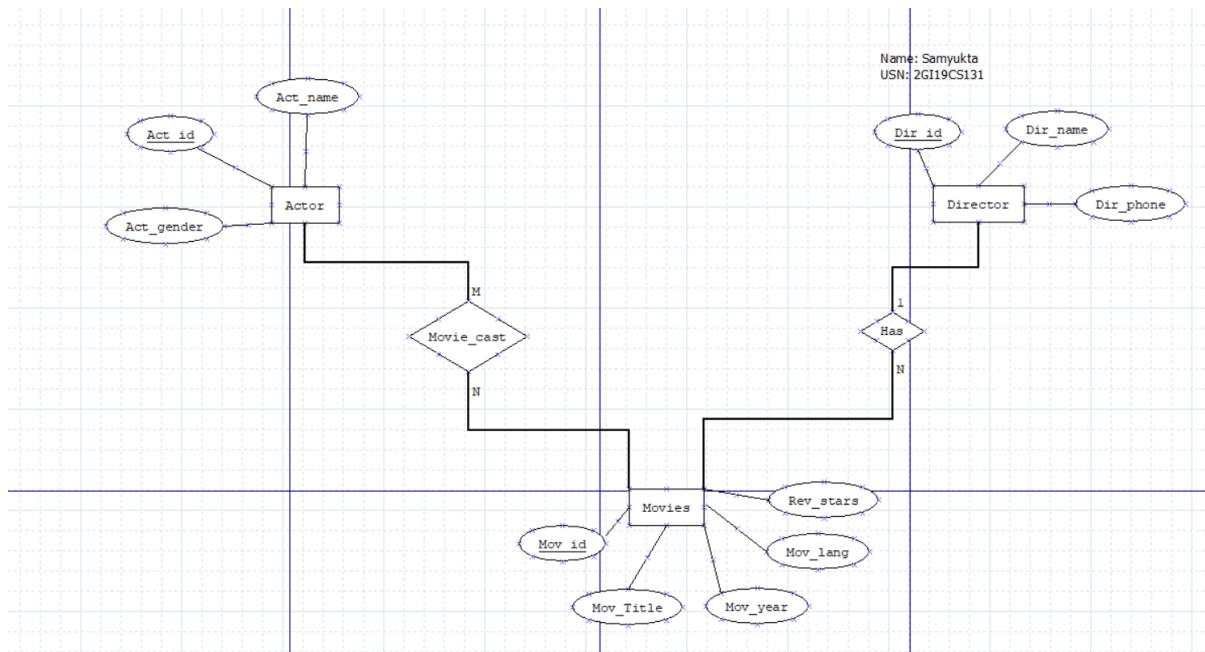
1. List the titles of all movies directed by ‘Sanjay Leela Bansali’.
2. Find the movie names where one or more actors acted in two or more movies.
3. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.
4. Update rating of all movies directed by ‘Ram Gopal Verma’ to 5.

Concepts: ER-Modelling is a first step towards building a database application. It helps in identifying various entities, their attributes and the relationship between them. The ER-Model helps the application developers to explain the customers, what all data would be stored and seek their suggestions to include all the data relevant to the application. While designing an ER-Model it is important to include only the attributes relevant to the entity types. Further after drawing ER-diagram the structural constraints, namely the cardinality ratios and participation constraints must be correctly indicated. There 7 mapping rules, which must be applied to the ER-diagram after completion to get the Relational model. The relational schema diagram for each relation must be drawn and the Primary key and foreign keys must be correctly indicated. Then the DDL statements must be used to create the tables in Oracle DBMS. Using INSERT command data must be inserted and using SQL queries the data must be checked for its correctness.

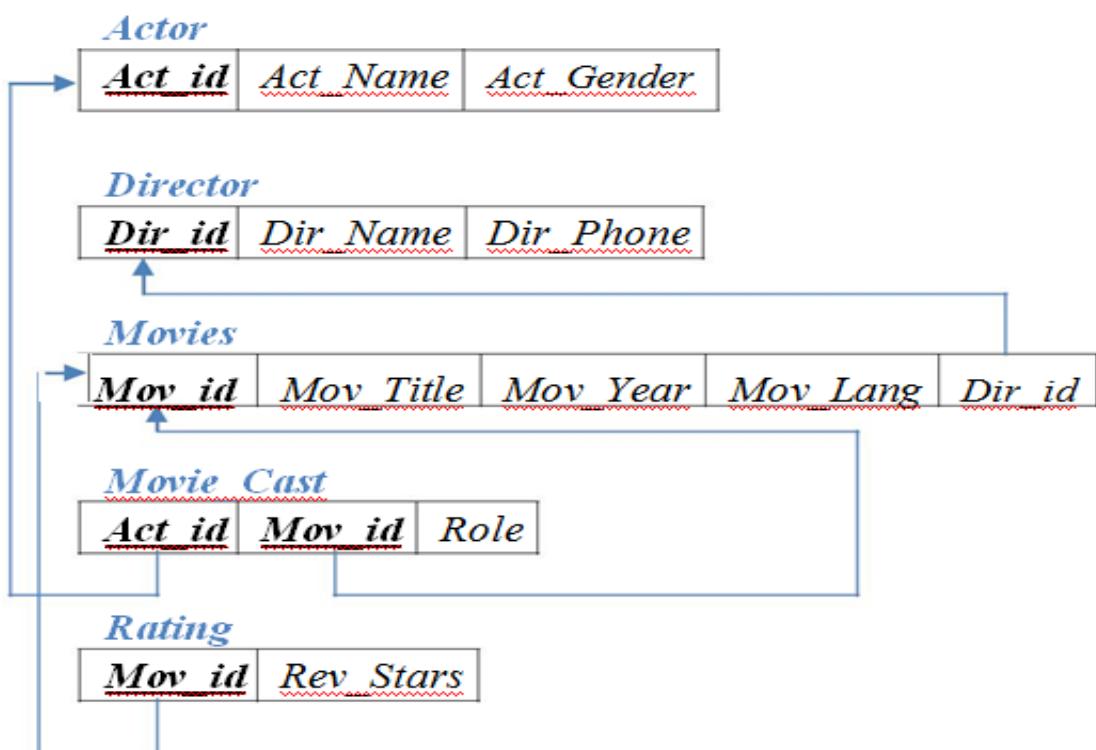
### Learning Outcomes:

- 33.ER-model is a set of concepts to describe data in graphical form.
34. There are 7 ER-to-Relational mapping rules to get Relational model from ER-Model.
35. Relational model is a set of concepts to describe data to RDBMS.
36. Relation, tuple, attribute, domain, Primary key, foreign key are the concepts in Relational mode.
37. DDL statements help us to create tables and specify constraints.
38. DML statements help us to populate and manipulate the database.
39. Design a GUI and write the program to connect to DBMS Server and display the data.
40. Learned to use SQL queries to list data stored in tables.

### ER Diagram:



### Schema Diagram:



### Create Table Statements:

```
CREATE  
TABLE  
ACTOR  
(  
    ACT_ID INTEGER PRIMARY KEY,  
    ACT_NAME VARCHAR(20),  
    ACT_GENDER CHAR(1));
```

Table created

```
SQL> DESC ACTOR;  
Name          Null?    Type  
---  
ACT_ID        NOT NULL NUMBER(38)  
ACT_NAME      VARCHAR2(20)  
ACT_GENDER    CHAR(1)
```

```
CREATE  
TABLE  
DIRECTOR  
(  
    DIR_ID INTEGER PRIMARY KEY,  
    DIR_NAME VARCHAR(20),  
    DIR_PHONE INTEGER);
```

Table created.

```
SQL> DESC DIRECTOR;
      Name          Null?    Type
-----+-----+-----+
  DIR_ID          NOT NULL NUMBER(38)
  DIR_NAME        VARCHAR2(20)
  DIR_PHONE       NUMBER(38)
```

```
CREATE
TABLE
MOVIES
(
    MOV_ID INTEGER PRIMARY KEY,
    MOV_TITLE VARCHAR(25),
    MOV_YEAR INTEGER,
    MOV_LANG VARCHAR(15),
    DIR_ID INTEGER,
    FOREIGN KEY (DIR_ID) REFERENCES
    DIRECTOR(DIR_ID));
```

Table created.

```
SQL> DESC MOVIES;
      Name          Null?    Type
-----+-----+-----+
  MOV_ID          NOT NULL NUMBER(38)
  MOV_TITLE        VARCHAR2(20)
  MOV_YEAR         NUMBER(38)
  MOV_LANG         VARCHAR2(15)
  DIR_ID           NUMBER(38)
```

```

CREATE
TABLE
MOVIE_CAST
(
    ACT_ID INTEGER,
    MOV_ID INTEGER,
    ROLE VARCHAR(10),
    PRIMARY KEY (ACT_ID,MOV_ID),
    FOREIGN KEY (ACT_ID) REFERENCES
    ACTOR(ACT_ID),
    FOREIGN KEY (MOV_ID) REFERENCES
    MOVIES(MOV_ID));

```

Table created.

Name	Null?	Type
ACT_ID	NOT NULL	NUMBER(38)
MOV_ID	NOT NULL	NUMBER(38)
ROLE		VARCHAR2(10)

```

CREATE
TABLE
RATING
(
    MOV_ID INTEGER PRIMARY KEY,
    REV_STARS VARCHAR(25),
    FOREIGN KEY (MOV_ID) REFERENCES
    MOVIES(MOV_ID));

```

Name	Null?	Type
MOV_ID	NOT NULL	NUMBER(38)
REV_STARS		VARCHAR2(25)

## INSERT Statements for each Table:

INSERT INTO ACTOR VALUES (101,'RADHIKA','F')

1 row created

INSERT INTO ACTOR VALUES (102,'ANKITHA','F');

1 row created

INSERT INTO ACTOR VALUES (103,'RADHIKA','F');

1 row created

INSERT INTO ACTOR VALUES (104,'CHETHAN','M');

1 row created

INSERT INTO ACTOR VALUES (105,'VIVAN','M');

1 row created

SQL> SELECT \* FROM ACTOR;

ACT_ID	ACT_NAME	A
101	RAHUL	M
102	ANKITHA	F
103	RADHIKA	F
104	CHETHAN	M
105	VIVAN	M

INSERT INTO DIRECTOR VALUES (201, ‘ANUP’, 918181818);

1 row created

INSERT INTO DIRECTOR VALUES (202, ‘SANJAU LEELA BANSALI’, 918181812);

1 row created

INSERT INTO DIRECTOR VALUES (203, ‘SHASHANK’, 918181813);

1 row created

```
INSERT INTO DIRECTOR VALUES (204, 'RAM GOPAL VERMA', 918181814);
```

1 row created

```
INSERT INTO DIRECTOR VALUES (205, 'ANAND', '918181815);
```

1 row created

```
SQL> SELECT * FROM DIRECTOR;
```

DIR_ID	DIR_NAME	DIR_PHONE
201	ANUP	918181818
202	SANJAY LEELA BANSALI	918181812
203	SHASHANK	918181813
204	RAM GOPAL VERMA	918181814
205	ANAND	918181815

```
INSERT INTO MOVIES VALUES (1001, 'MANASU', 2017, 'KANNADA', 201);
```

1 row created.

```
INSERT INTO MOVIE VALUES (1002, 'AAKASHAM', 2015, 'TELUGU', 202);
```

1 row created.

```
INSERT INTO MOVIE VALUES (1003, 'KALIYONA', 2008, 'KANNADA', 201);
```

1 row created.

```
INSERT INTO MOVIE VALUES (1004, 'WAR HOUSE', 2011, 'ENGLISH', 204);
```

1 row created.

```
INSERT INTO MOVIE VALUES (1005, 'HOME', 2012, 'ENGLISH', 205);
```

1 row created.

```
SQL> SELECT * FROM MOVIES;
```

MOV_ID	MOV_TITLE	MOV_YEAR	MOV_LANG	DIR_ID
1001	MANASU	2017	KANNADA	201
1002	AAKASHAM	2015	TELUGU	202
1003	KALIYONA	2008	KANNADA	201
1004	WAR HORSE	2011	ENGLISH	204
1005	HOME	2012	ENGLISH	205

INSERT INTO MOVIE\_CAST VALUES (101, 1002, 'HERO');

1 row created.

INSERT INTO MOVIE\_CAST VALUES (101, 1001, 'HERO');

1 row created.

INSERT INTO MOVIE\_CAST VALUES (103, 1003, 'HERIONE');

1 row created.

INSERT INTO MOVIE\_CAST VALUES (103, 1002, 'GUEST');

1 row created.

INSERT INTO MOVIE\_CAST VALUES (104, 1004, 'HERO');

1 row created.

```
SQL> SELECT * FROM MOVIE_CAST;
```

ACT_ID	MOV_ID	ROLE
101	1002	HERO
101	1001	HERO
103	1003	HERIONE
103	1002	GUEST
104	1004	HERO

```
INSERT INTO RATING VALUES (1001, 4);
```

1 row created.

```
INSERT INTO RATING VALUES (1002, 2);
```

1 row created.

```
INSERT INTO RATING VALUES (1003, 5);
```

1 row created.

```
INSERT INTO RATING VALUES (1004, 5);
```

1 row created.

```
INSERT INTO RATING VALUES (1005, 3);
```

1 row created.

```
SQL> SELECT * FROM RATING;  
  
MOV_ID REV_STARS  
-----  
1001 4  
1002 2  
1003 5  
1004 5  
1005 3
```

### SQL Queries:

1. List the titles of all movies directed by ‘Sanjay Leela Bansali.

```
SELECT MOV_TITLE
```

```
FROM MOVIES
```

```
WHERE DIR_ID IN (SELECT DIR_ID
```

```
FROM DIRECTOR
```

```
WHERE DIR_NAME = ‘Sanjay Leela Bansali’);
```

```
MOV_TITLE
```

```
-----
```

```
AAKASHAM
```

```
-----
```

```
SQL>
```

2. Find the movie names where one or more actors acted in two or more movies.

```
SELECT MOV_TITLE  
FROM MOVIES M, MOVIE_CAST MV  
WHERE M.MOV_ID=MV.MOV_ID AND ACT_ID IN (SELECT ACT_ID  
FROM MOVIE_CAST GROUP BY ACT_ID HAVING COUNT  
(ACT_ID)>1)  
GROUP BY MOV_TITLE  
HAVING COUNT (*)>1;
```

```
MOV_TITLE
```

```
-----
```

```
AAKASHAM
```

```
-----
```

```
SQL>
```

3. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.

```
SELECT MOV_TITLE, MAX (REV_STARS)
FROM MOVIES
INNER JOIN RATING USING (MOV_ID)
GROUP BY MOV_TITLE
HAVING MAX (REV_STARS)>0
ORDER BY MOV_TITLE;
```

MOV_TITLE	MAX(REV_STARS)
AAKASHAM	2
HOME	3
KALIYONA	5
MANASU	4
WAR HORSE	5

4. Update rating of all movies directed by ‘Ram Gopal Verma’ to 5.

```
UPDATE RATING SET REV_STARS=5
WHERE MOV_ID IN (SELECT MOV_ID FROM MOVIES
WHERE DIR_ID IN (SELECT DIR_ID
FROM DIRECTOR
WHERE DIR_NAME = ‘RAM GOPAL VERMA’));
```

## Conclusion:

We learnt to use the open-source ER-design tool dia and created the ER-Model for the above said problem statement. Converted the ER-diagram into relational schema diagram by applying the ER- to relational rules. We identified primary and foreign keys and created all the relations in Oracle DBMS using DDL statements. Further, the database was populated with real data using insert statements. The content of each table was displayed using SELECT sql statement. We learnt about update and delete sql statements and also learnt about alter table command to modify or add constraints to the table structure after they are created.