ESTD. 1939

KARNATAK LAW SOCIETY'S
GOGTE INSTITUTE OF TECHNOLOGY
"JNANA GANGA" UDYAMBAG, BELAGAVI-590008,
KARNATAKA, INDIA.
Approved by AICTE & UGC
Permanently Affiliated and Autonomous Institution Under
Visvesvaraya Technological University, Belagavi
www.git.edu

ESTD. 1979

DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

DATABASE APPLICATION LABORATORY MANUAL

18CSL48

4th SEMESTER
2019 – 2020

Course learning objectives

1. Gain a good understanding of the architecture and functioning of Database Management Systems as well as associated tools and techniques.
2. Understand and apply the principles of data modeling using Entity Relationship and develop a good database design.
3. Apply Normalization techniques to normalize a database.
4. Understand the use of Structured Query Language (SQL) and its syntax.
5. Learn the tools required for graphical user interface design .

Course Outcome (COs)

At the end of the course, the student will be able to		Bloom's Level
1	**Apply** the ER-Modeling concepts, Normalization and design a database accordingly	**L3**
2	**Demonstrate** use of DDL and DML statements	**L3**
3	**Identify and write** SQL statements for the given end user queries	**L3**
4	**Demonstrate** the use of GUI tools	**L3**

Text Books:

1. Elmasri and Navathe: Fundamentals of Database Systems, Addison-Wesley, 3rd edition and onwards.
2. Raghu Ramakrishnan and Johannes Gehrke: Database Management Systems, McGraw-Hill, 2nd edition and onwards.

Reference Books:

1. Silberschatz, Korth and Sudharshan: Data base System Concepts, Mc-GrawHill, 3rd edition and onwards.
2. C.J. Date, A. Kannan, S. Swamynatham: A Introduction to Database Systems, Pearson education, 5th edition and onwards.

Table of Contents

INTRODUCTION TO SQL

Pronounced as SEQUEL: Structured English QUERY Language

- Pure non-procedural query language
- Designed and developed by IBM, Implemented by Oracle
- 1978 System/R IBM- 1st Relational DBMS
- 1979 Oracle and Ingres
- 1982 SQL/DS and DB2 IBM
- Accepted by both ANSI + ISO as Standard Query Language for any RDBMS
- SQL86 (SQL1) : first by ANSI and ratified by ISO (SQL-87), minor revision on 89
- (SQL-89)
- SQL92 (SQL2) : major revision
- SQL99 (SQL3) : add recursive query, trigger, some OO features, and non-scholar type
- SQL2003 : XML, Window functions, and sequences (Not free)
- Supports all the three sublanguages of DBMS: DDL, DML, DCL
- Supports Aggregate functions, String Manipulation functions, Set theory operations,
- Date Manipulation functions, rich set of operators (IN, BETWEEN, LIKE, IS NULL,
- EXISTS)
- Supports REPORT writing features and Forms for designing GUI based applications

DATA DEFINITION, CONSTRAINTS, AND SCHEMA CHANGES

Used to CREATE, ALTER, and DROP the descriptions of the database tables (relations)

Data Definition in SQL

CREATE, ALTER and DROP

table……relation

row……..…….tuple

column…….……attribute

DATA TYPES

- Numeric: NUMBER, NUMBER(s,p), INTEGER, INT, FLOAT, DECIMAL
- Character: CHAR(n), VARCHAR(n), VARCHAR2(n), CHAR VARYING(n)
- Bit String: BLOB, CLOB
- Boolean: true, false, and null
- Date and Time: DATE (YYYY-MM-DD) TIME(HH:MM:SS)
- Timestamp: DATE + TIME
- USER Defined types

CREATE SCHEMA

Specifies a new database schema by giving it a name

Ex: CREATE SCHEMA COMPANY AUTHORIZATION Jsmith;

CREATE TABLE

- Specifies a new base relation by giving it a name, and specifying each of its attributes andtheir data types

Syntax of CREATE Command:

CREATE TABLE<table name>(<Attribute A1><Data Type D1> [<Constarints>],

<Attribute A2><Data Type D2> [<Constarints>],

…….

<Attribute An><Data Type Dn> [<Constarints>],

[<integrity-constraint1>, <integrity-constraint k>]);

- A constraint NOT NULL may be specified on an attribute

A constraint NOT NULL may be specified on an attribute

Ex: CREATE TABLE DEPARTMENT (

DNAME VARCHAR(10) NOT NULL,

DNUMBER INTEGER NOT NULL,

MGRSSN CHAR(9), MGRSTARTDATE CHAR(9));

- Specifying the unique, primary key attributes, secondary keys, and referential integrityconstraints (foreign keys).

Ex: CREATE TABLE DEPT (

DNAME VARCHAR(10) NOT NULL,

DNUMBER INTEGER NOT NULL,

MGRSSN CHAR(9),

MGRSTARTDATE CHAR(9),

PRIMARY KEY (DNUMBER),

UNIQUE (DNAME),

FOREIGN KEY (MGRSSN) REFERENCES EMP(SSN));

- We can specify RESTRICT, CASCADE, SET NULL or SET DEFAULT on referentialintegrity constraints (foreign keys)

Ex: CREATE TABLE DEPT (

DNAME VARCHAR(10) NOT NULL,

DNUMBER INTEGER NOT NULL,

MGRSSN CHAR(9), MGRSTARTDATE CHAR(9),

PRIMARY KEY (DNUMBER),

UNIQUE (DNAME),

FOREIGN KEY (MGRSSN) REFERENCES EMP

ON DELETE SET DEFAULT ON UPDATE CASCADE);

DROP TABLE

- Used to remove a relation (base table) and its definition.
- The relation can no longer be used in queries, updates, or any other commands since itsdescription no longer exists

Example: DROP TABLE DEPENDENT;

ALTER TABLE

- Used to add an attribute to/from one of the base relations drop constraint -- The newattribute will have NULLs in all the tuples of the relation right after the command isexecuted; hence, the NOT NULL constraint is not allowed for such an attribute.
  Example: ALTER TABLE EMPLOYEE ADD JOB VARCHAR2 (12);
- The database users must still enter a value for the new attribute JOB for eachEMPLOYEE tuple. This can be done using the UPDATE command.

DROP A COLUMN (AN ATTRIBUTE)

- ALTER TABLE COMPANY.EMPLOYEE DROP ADDRESS CASCADE; Allconstraints and views that reference the column are dropped automatically, along with thecolumn. ALTER TABLE COMPANY.EMPLOYEE DROP ADDRESS RESTRICT;Successful if no views or constraints reference the column. ALTER TABLE

COMPANY.DEPARTMENT ALTER MGRSSN DROP DEFAULT;

- ALTER TABLE COMPANY.DEPARTMENT ALTER MGRSSN SET DEFAULT 333445555;

BASIC QUERIES IN SQL

- SQL has one basic statement for retrieving information from a database; the SLELECTstatement
- This is not the same as the SELECT operation of the relational algebra
- Important distinction between SQL and the formal relational model;
- SQL allows a table (relation) to have two or more tuples that are identical in all their
- attribute values
- Hence, an SQL relation (table) is a multi-set (sometimes called a bag) of tuples; it is not aset of tuples
- SQL relations can be constrained to be sets by using the CREATE UNIQUE INDEXcommand, or by using the DISTINCT option
- Basic form of the SQL SELECT statement is called a mapping of a SELECT-FROMWHERE block

  SELECT <attribute list> FROM <table list> WHERE <condition>
- <attribute list> is a list of attribute names whose values are to be retrieved by the query
- <table list > is a list of the relation names required to process the query
- <condition> is a conditional (Boolean) expression that identifies the tuples to be retrievedby the query

EXPERIMENT - 1 : Database Design – ER Modeling

Problem Definition: 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 a 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)

Objectives of the Experiment:

1. To introduce the ER-Modeling concepts
2. Introduce an ER-Modeling tool to design an ER-Model for a given application
3. Present ER-Mapping Rules and translate ER-Model to Relational Model
4. Introduce SQL - DDL and DML commands to construct a relational database

Basic Concepts of ER Model/SQL/DDL/DML as applicable

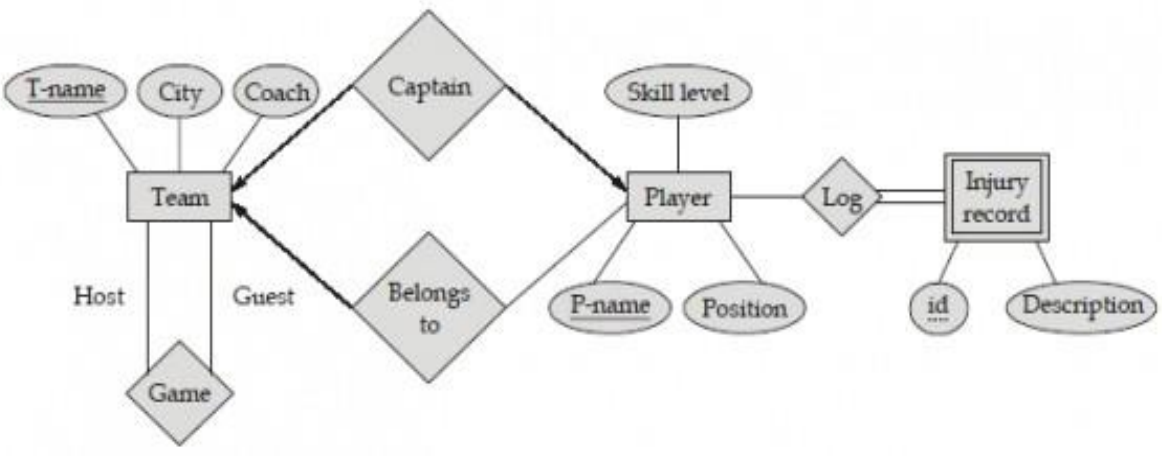
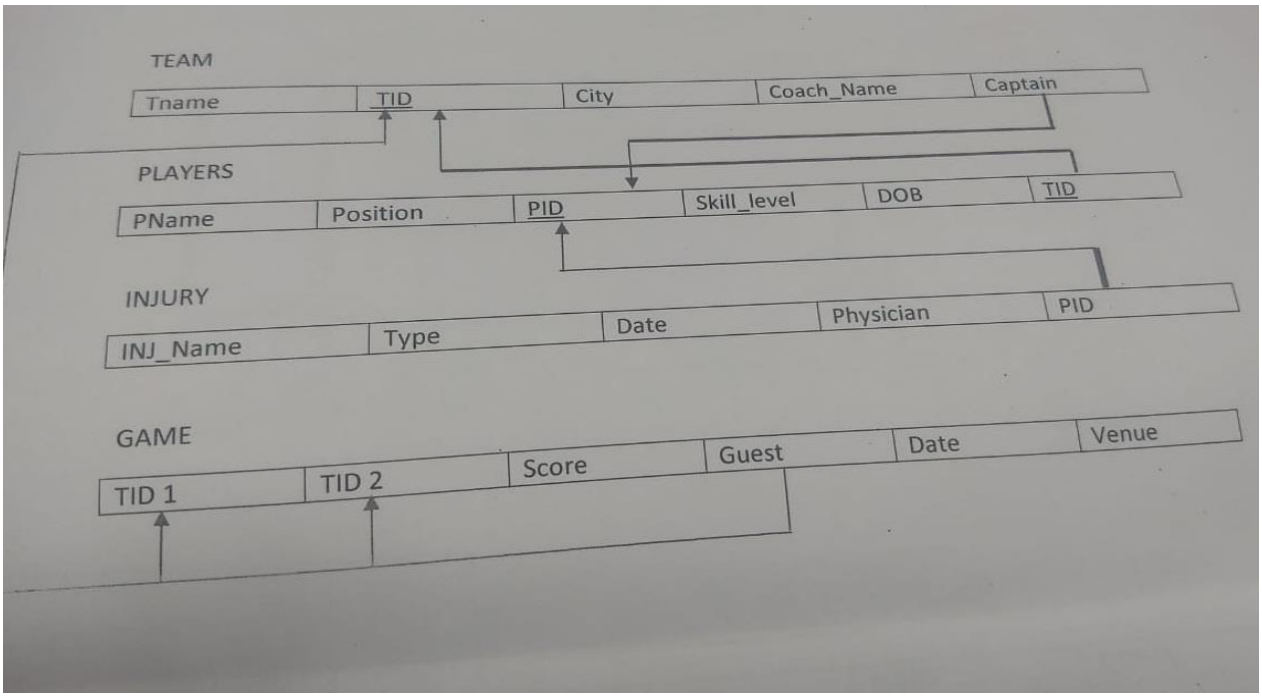


Fig.1: ER Diagram for National Hockey League

Relational Schema Diagram



Procedure for Creating tables:

1. create table team till coach name

```
create table team
(
team_idint,
team_namevarchar(10),
cityvarchar(10),
coach_namevarchar(10),
primary key(team_id)
);
```

2. create table players with all attributes

```
create table players
(
pidint,
p_namevachar(10),
positionvarchar(10),
skillvarchar(10),
dob date,
team_idint,
primary key(pid),
```

```
foreign key(team_id) references team(team_id)
);
```

3. create table injury

```
create table injury
(
inj_namevarchar(10),
pidint,
I_date date,
physicianvarchar(10),
typevarchar(10),
primary key(inj_name,pid),
foreign key (pid) references player(pid)
);
```

4. create table games

```
create table games
(
tid1int,
tid2int,
scorevarchar(5),
guestint,
g_date date,
venuevarchar(10),
primary key(tid1,tid2,guest),
foreign key(tid1) references team(team_id),
foreign key(tid2) references team(team_id)
foreign key(guest) references team(team_id)
);
```

5. insert values in team table..

6. insert values in players table.

7. addcaptain_id to team table

```
alter table team

addcaptain_idint;
```

8. update table team set captain_id=NULL;

9. makecaptain_id as foreign key

alter table team

add foreign(key) references player(pid);

10. update table team

setcaptain_id=101

whereteam_id=1;

Learning Outcome of the Experiment

At the end of the session, students should be able to :

1. Design an ER-Model for a given application scenario [L5, CO 2, PO1]
2. Demonstrate the use of ER-Modelling tool to construct the models [L3, CO 2, PO3]
3. Design the Relational schema by applying Mapping Rules and analyse its correctness [L4, CO 2, PO3]
4. Construct the database and Demonstrate the execution of Queries. [L5, CO 2, PO4]

Conclusions: The students learned the procedure to identify the Entities, the relationships from the given problem statement and successfully designed the ER-Schema using a software tool. Further, they applied the ER-to-Relational mapping to get the final Relational Schema. The entire Database complete in all respects is then used to create the database in Oracle 10g, populate them and test some queries.

EXPERIMENT 2: Database Design – ER Modeling

Problem Definition: Design an ER-Model for an educational institute which is required to record the students attendance and IA performance in all the subjects and inform the same to their parents. The institute will have many departments, each with their own faculty and Head of the department. The subjects the students study could be core or elective. A faculty has to take atleast one subject and atmost two subjects and the subjects are not shared. Every student takes three subjects and the final marks in each subject is the average of the best two of the three tests. Map the ER-Model to relational schema using the seven mapping rules and construct the database in Oracle DBMS.

Objectives of the Experiment:

1. To introduce the ER-Modeling concepts
2. Introduce an ER-Modeling tool to design an ER-Model for a given application
3. Present ER-Mapping Rules and translate ER-Model to Relational Model
4. Introduce SQL - DDL and DML commands to construct a relational database

Basic Concepts of ERModel/SQL/DDL/DML as applicable

ER-SCHEMA

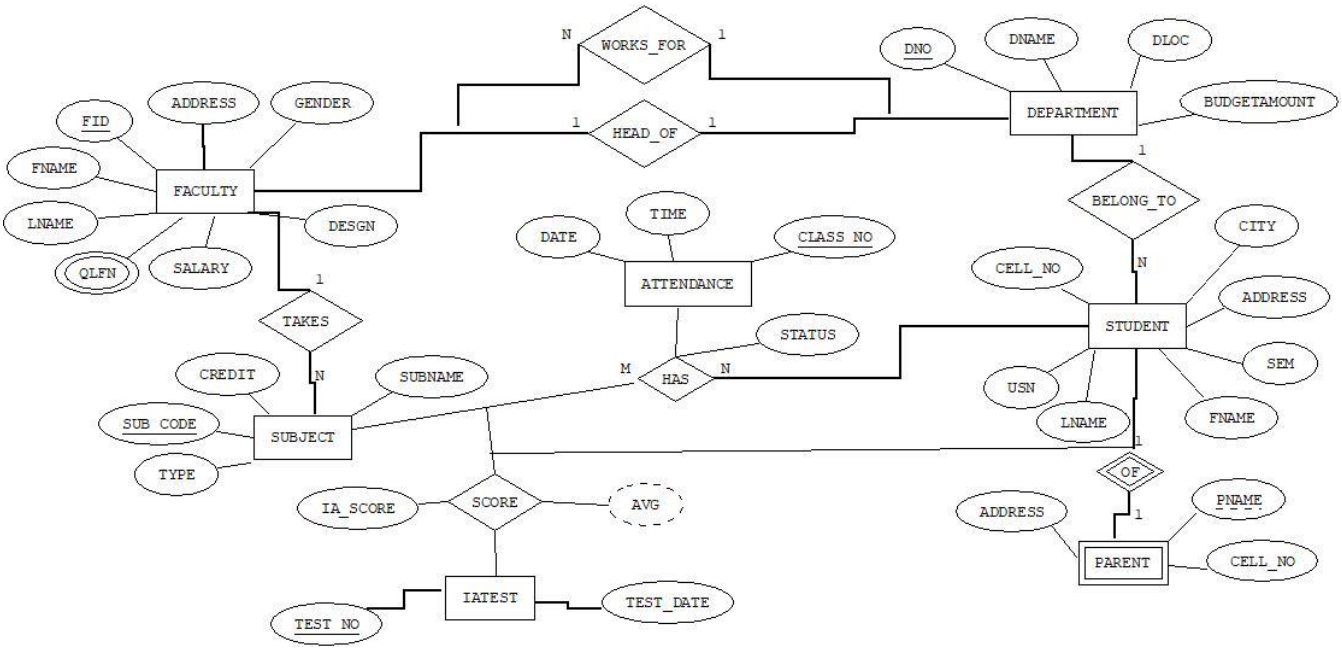
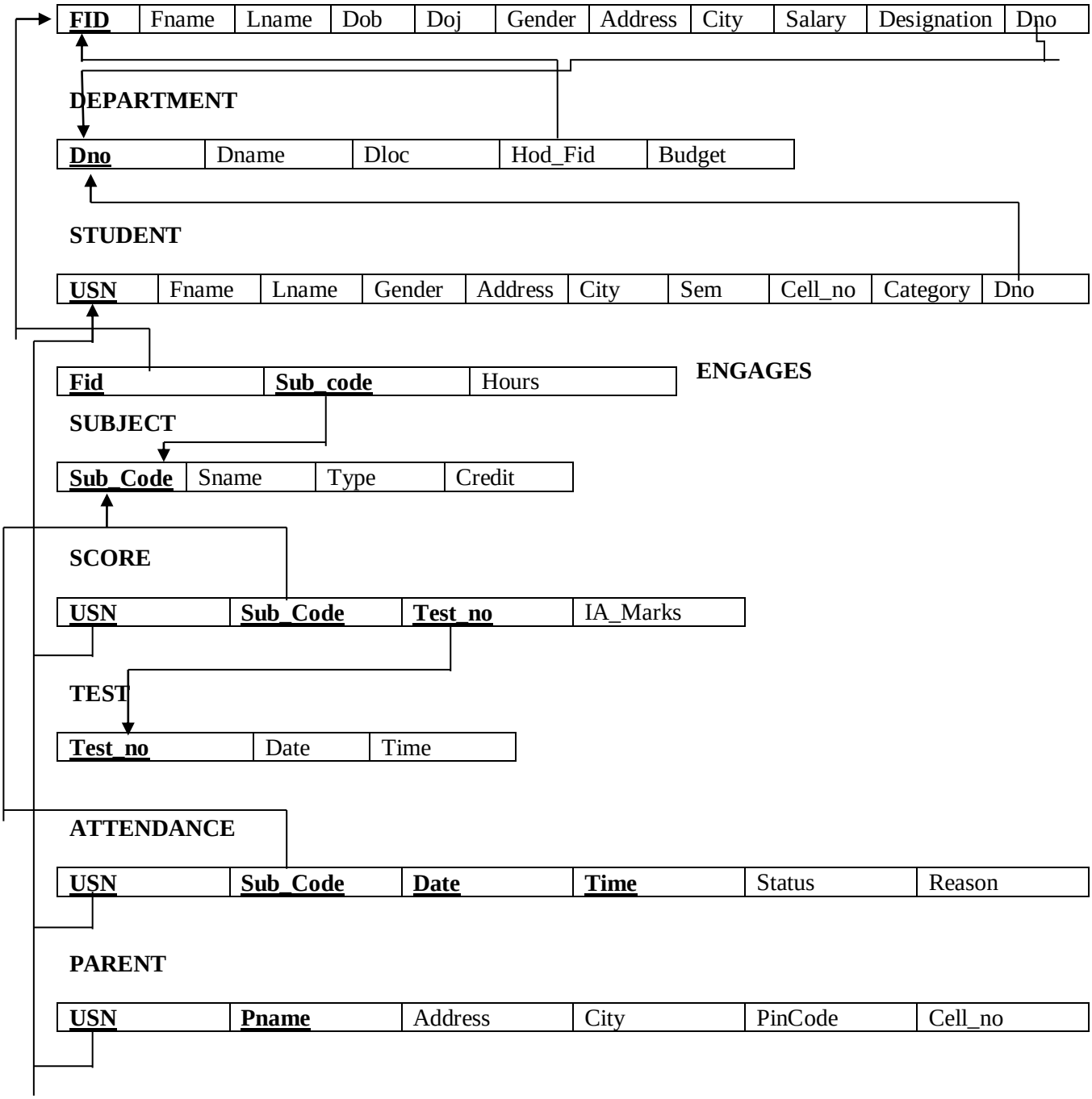


Fig. 2 ER-Schema of Institute Database Application

SCHEMA Diagram of INSTITUTE Database Application



CREATE TABLE

1. CREATE TABLE STUDENT(USN CHAR(10), Fnamevarchar(15), Lnamevarchar(15), Gender char(1), Address varchar(30), City varchar(15), semint, cell_no char(10), dnoint, Primary key(USN), Foreign key(Dno) references DEPARTMENT(Dno));
2. CREATE TABLE DEPARTMENT(Dnoint, Dnamevarchar(20), Dlocvarchar(20), Head_Fid integer, PRIMARY KEY(Dno), FOREIGN KEY(Head_Fid) references FACULTY(Fid));

INSERT STATEMENT

1. INSERT INTO STUDENT values('2GI16CS001','Ajay','Kulkarni','M','Tilakwadi','Belgaum',3,'9399392921',2);
2. INSERT INTO DEPARTMENT values(&Dno, '&Dname','&Dloc',&Head_Fid);

UPDATE STATEMENT

1. Update FACULTY set salary=&new_salary where Fid=&Fid;

SQL QUERIES

1. SELECT F_name, L_name, Dname from STUDENT S, DEPARTMENT D WHERE S.Dno=D.Dno and Dname='CSE';

Learning Outcome of the Experiment

At the end of the session, students should be able to :

1. Design an ER-Model for a given application scenario [L5, CO 2, PO1]
2. Demonstrate the use of ER-Modelling tool to construct the models [L3, CO 2, PO3]
3. Design the Relational schema by applying Mapping Rules and analyse its correctness [L4, CO 2, PO3]
4. Construct the database and Demonstrate the execution of Queries. [L5, CO 2, PO4]

Conclusions : The students learned the procedure to identify the Entities, the relationships from the given problem statement and successfully designed the ER-Schema using a software tool. Further, they applied the ER-to-Relational mapping to get the final Relational Schema. The entire Database complete in all respects is then used to create the database in Oracle 10g, populate them and test some queries.

EXPERIMENT – 3: AIRLINE DATABASE

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(Atleast 5 records in each table) for the given database.

Write SQL queries to

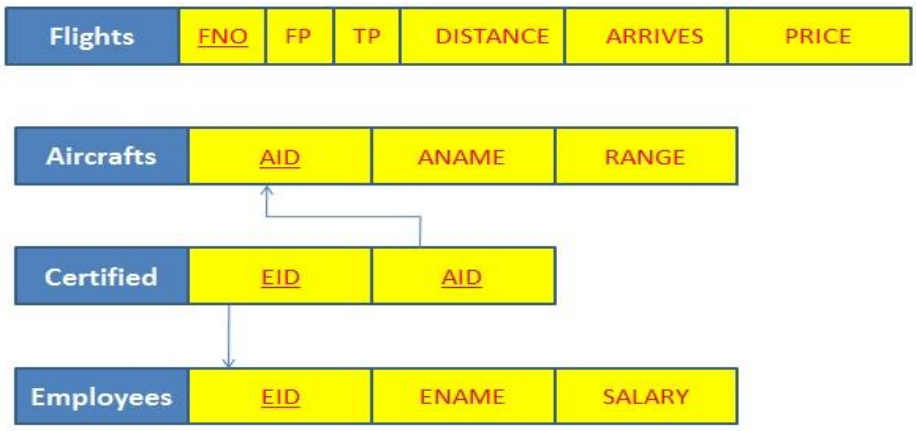
1. Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80,000.

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 aircraft that can be used on routes from Bengaluru to New Delhi

SCHEMA DIAGRAM



CREATE TABLE STATEMENTS:

```
CREATE TABLE FLIGHTS
 (no integer,
fromPlacevarchar(20),
toPlacevarchar(20),
distance integer,
departs date,
arrives date,
price real,
 PRIMARY KEY(no));
```

```
CREATE TABLE AIRCRAFT
 (aid integer,
anamevarchar(15),
cruisingrange integer,
  PRIMARY KEY(aid));
```

```
CREATE TABLE EMPLOYEES
 (eid integer,
enamevarchar(15),
salary integer,
PRIMARY KEY(eid));
```

SQL)**SELECT * FROM FLIGHTS;**

NO	FROMPLACE	TOPLACE	DISTANCE	DEPARTS	ARRIVES	PRICE
255	bangalore	frankfurt	200	01-AUG-11	01-AUG-11	5000
256	bangalore	frankfurt	200	01-AUG-11	01-AUG-11	8000
257	bangalore	delhi	200	01-AUG-11	01-AUG-11	5000
258	bangalore	delhi	200	01-AUG-11	01-AUG-11	6000
259	bangalore	mangalore	200	01-AUG-11	01-AUG-11	8000

```
SQL> SELECT * FROM AIRCRAFT;

    AID ANAME      GRANGE
------- ---------- ----------
    685 boeing15         1000
    686 boeing10         2000
    687 skytrain         1000
    688 avenger           100
```

```
SQL> SELECT * FROM EMPLOYEES;

    EID ENAME      SALARY
------- ---------- ----------
    101 asha            90000
    102 arun            85000
    103 anand            3000
    104 ramya            4000
```

```
SQL> SELECT * FROM CERTIFIED;
    EID        AID
-------- ---------
    101        685
    101        686
    101        687
    101        688
    102        685
    103        686
    103        687
```

QUERIES:

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

```
SELECT DISTINCT A.aname
 FROM    AIRCRAFT A,   CERTIFIED  C, EMPLOYEE E
```

```
WHERE  A.aid = C.aid  AND  C.eid =E.eid AND E.salary>80000;
```

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

```
SELECT   C.eid,   MAX(a.cruisingrange)
FROM    CERTIFIED  C,  AIRCRAFT A
WHERE   C.aid = A.aid
GROUP BY  C.eid
HAVING    COUNT(*) > 3;
```

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

```
SELECT  E.ename

FROM EMPLOYEES  E

WHERE  E.salary<   (SELECT min(price)

      FROM FLIGHTS

            WHERE   fromPlace = 'bangalore'  ANDtoPlace = 'frankfurt');
```

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

```
SELECT    A.aid

FROM   AIRCRAFT A

WHERE    A.cruisingrange> (SELECT min(distance)

                  FROM FLIGHTS

             WHERE  fromPlace='bangalore'   AND    toPlace = 'delhi');
```

Learning Outcome of the Experiment

At the end of the session, students should be able to :

1. Design a Schema Diagram for a given application scenario [L4, CO 2, PO3]
3. Construct the database and Demonstrate the execution of Queries. [L5, CO 2, PO4]

Conclusions : The students learned the procedure to map the given scenerio to get the final Relational Schema. The entire Database complete in all respects is then used to create the database in Oracle 10g, populate them and test some queries.

EXPERIMENT – 4 : ORDER DATABASE

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,Saleman_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 salesmen names and customer names for whom order amount is more than 4000. (Use UNION operation.)

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

SCHEMA DIAGRAM

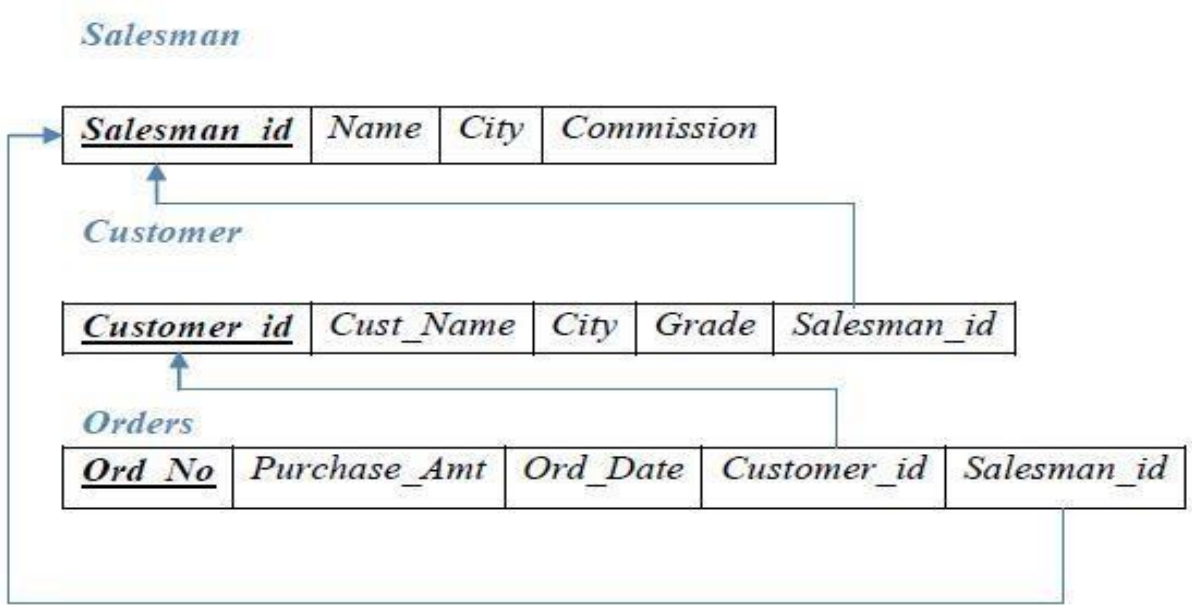


Table Creation:

```
REATE TABLE SALESMAN(
SALESMAN_ID INT PRIMARY KEY,
NAME VARCHAR(10),
CITY VARCHAR (15), COMMISSION INT);

CREATE TABLE CUSTOMER (
CUSTOMER_ID INT PRIMARY KEY,
CUST_NAME VARCHAR (10),
CITY VARCHAR (10),
GRADE INT,
SALESMAN_ID     INT,
FOREIGN KEY(SALESMAN_ID) REFERENCES SALESMAN(SALESMAN_ID) ON DELETE
SET NULL);


CREATE TABLE ORDERS (
ORD_NO INT PRIMARY KEY,
PURCHASE_AMT INT,
 ORD_DATE DATE,
CUSTOMER_ID      INT,
SALESMAN_ID      INT,
FOREIGN KEY(CUSTOMER_ID) REFERENCES CUSTOMER (CUSTOMER_ID)
ON DELETE CASCADE,
FOREIGN KEY(SALESMAN_ID) REFERENCES SALESMAN(SALESMAN_ID)
ON DELETE CASCADE);
```

Values for tables

SQL> INSERT INTO SALESMAN **VALUES**(&SALESMAN_ID,'&NAME','&CITY',&COMMISSION);

SQL> INSERT INTO CUSTOMER

VALUES(&CUSTOMER_ID,'&CUST_NAME','&CITY','&GRADE',&SALESMAN_ID);

SQL> INSERT INTO ORDERS

VALUES(&ORD_NO,&PURCHASE_AMT,'&ORD_DATE',&CUSTOMER_ID,&SALESMAN_ID);

SELECT*FROM SALESMAN;

SALESMAN_ID	NAME	CITY	COMMISSION
1000	RAJ	BENGALURU	50
2000	ASHWIN	TUMKUR	30
3000	BINDU	MUMBAI	40
4000	LAVANYA	BENGALURU	40
5000	ROHIT	MYSORE	60

SELECT*FROM CUSTOMER;

CUSTOMER_ID	CUST_NAME	CITY	GRADE	SALESMAN_ID
11	INFOSYS	BENGALURU	5	1000
22	TCS	BENGALURU	4	2000
33	WIPRO	MYSORE	7	1000
44	TCS	MYSORE	6	2000
55	ORACLE	TUMKUR	3	3000

SELECT*FROM ORDERS;

ORD_NO	PURCHASE_AMT	ORD_DATE	CUSTOMER_ID	SALESMAN_ID
1	200000	12-APR-16	11	1000
2	300000	12-APR-16	11	2000
3	400000	15-APR-17	22	1000

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

```
SELECT
COUNT(CUSTOMER_ID)FROM
CUSTOMER

WHERE GRADE>(SELECT AVG(GRADE)

            FROM CUSTOMER

            WHERE CITY ='BENGALURU');
```

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

3
```

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

```
SELECT
S.NAME,COUNT(CUSTOMER_ID)FROM
SALESMAN S, CUSTOMER C

WHERE S.SALESMAN_ID=C.SALESMAN_ID
GROUP BY S.NAME

HAVING COUNT(CUSTOMER_ID)>1;
```

```
NAME            COUNT(CUSTOMER_ID)

---------- ------------------ ASHWIN          2
RAJ                                          2
```

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

```
SELECT S.NAME, C.NAME FROM
SALESMAN, CUSTOMER C, ORDER
O

WHERE
S.SALESMAN_ID=O.SALESMAN_ID
AND
C.CUSTOMER_ID=O.CUSTOMER_ID
AND

O.PURCHASE_AMT>4000;
```

4. Demonstrate the DELETE operation by removing salesman withid1000. All his orders must also be deleted.

DELETE from salesman

WHERE salesman_id = 1000;

1 row deleted.

SELECT*__FROM__SALESMAN;

SALESMAN_ID	NAME	CITY	COMMISSION
2000	ASHWIN	TUMKUR	30
3000	BINDU	MUMBAI	40
4000	LAVANYA	BENGALURU	40
5000	ROHIT	MYSORE	60

SELECT*__FROM__ CUSTOMER;

CUSTOMER_ID	CUST_NAME	CITY	GRADE	SALESMAN_ID
11	INFOSYS	BENGALURU	5	
22	TCS	BENGALURU	4	2000
33	WIPRO	MYSORE	7	
44	TCS	MYSORE	6	2000
55	ORACLE	TUMKUR	3	3000

SELECT*__FROM__ ORDERS;

ORD_NO	PURCHASE_AMT	ORD_DATE	CUSTOMER_ID	SALESMAN_ID
2	300000	12-APR-16	11	2000

Learning Outcome of the Experiment

At the end of the session, students should be able to :

2. Design a Schema Diagram for a given application scenario [L4, CO 2, PO3]
4. Construct the database and Demonstrate the execution of Queries. [L5, CO 2, PO4]

Conclusions : The students learned the procedure to map the given scenerio to get the final Relational Schema. The entire Database complete in all respects is then used to create the database in Oracle 10g, populate them and test some queries.

EXPERIMENT 5: MOVIE DATABASE

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)

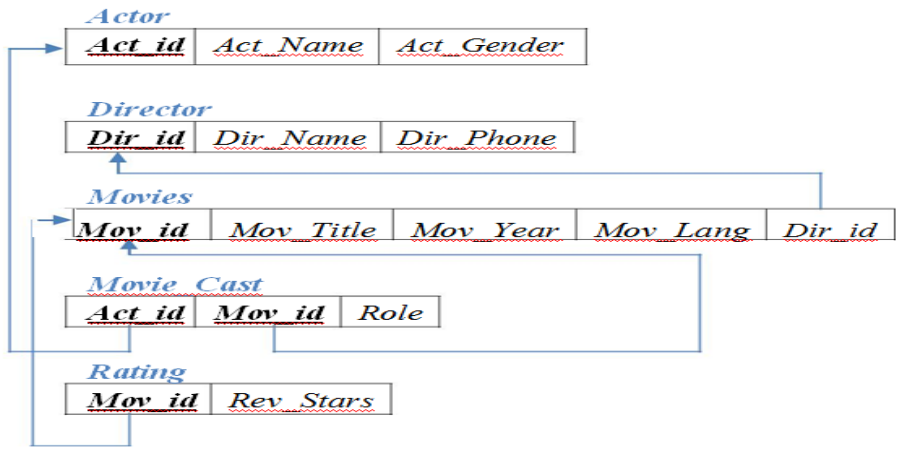
MOVIE_CAST (Act_id,Mov_id, Role)

RATING (Mov_id, Rev_Stars)

Write SQL queries to

1. **List the titles of all movies directed by 'Sanjay LeelaBansali'.**
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 GopalVerma'to 5.**

Solution:



CREATE TABLE STATEMENTS

```
CREATE TABLE ACTOR
 (
        ACT_ID NUMBER (3) PRIMARY KEY,
        ACT_NAME  VARCHAR (20),
        ACT_GENDER CHAR (1),
);
CREATE TABLE DIRECTOR
(
        DIR_ID NUMBER (3) PRIMARY KEY,
        DIR_NAME VARCHAR (20),
        DIR_PHONE NUMBER (10),
);
CREATE TABLE MOVIES
(
        MOV_ID NUMBER (4),
        MOV_TITLE VARCHAR (25),
        MOV_YEAR NUMBER (4),
        MOV_LANG VARCHAR (12),
        DIR_ID NUMBER (3),
        PRIMARY KEY (MOV_ID),
        FOREIGN KEY (DIR_ID) REFERENCES DIRECTOR (DIR_ID)
);
CREATE TABLE MOVIE_CAST
(
        ACT_ID NUMBER (3),
        MOV_ID NUMBER (4),
        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)
);
CREATE TABLE RATING
 (
        MOV_ID NUMBER (4),
```

```
    REV_STARS VARCHAR (25),
    PRIMARY KEY (MOV_ID),
    FOREIGN KEY (MOV_ID) REFERENCES MOVIES (MOV_ID)
);
```

QUERIES

1. List the titles of all movies directed by 'Sanjay LeelaBansali'.

```
SELECT MOV_TITLE
FROM MOVIES
WHERE DIR_ID IN (SELECT DIR_ID
FROM DIRECTOR
WHERE DIR_NAME = _SANJAYLELABANSALI');
```

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;
```

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;
```

4. Update rating of all movies directed by 'Ram GopalVerma' 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  =  _STEVEN
SPIELBERG‘));
```

Learning Outcome of the Experiment

At the end of the session, students should be able to :

3. Design a Schema Diagram for a given application scenario [L4, CO 2, PO3]
5. Construct the database and Demonstrate the execution of Queries. [L5, CO 2, PO4]

Conclusions : The students learned the procedure to map the given scenerio to get the final Relational Schema. The entire Database complete in all respects is then used to create the database in Oracle 10g, populate them and test some queries.

Mapping of COs and POs

Course Outcomes	Activities (Mention the S.No.)	Program Outcomes											
		1	2	3	4	5	6	7	8	9	10	11	12
CO-1	1,2,3	L	L	-			-	-	M	-	-	-	-
CO-2	1,2,3	M	L	-		-	-	-	L	-	-	-	-
CO-3	1,2,3	M	L	-	-	-	-	-	L	-	-	-	-
CO-4	1,2,3	M	L	-			-	-	M	-	-	-	-
CO-5	4		M										

Mapping of COs and PSOs

Course Outcomes	Activities (Mention the S.No.)	Program Specific Outcomes			
		PSO1	PSO2	PSO3	…
CO-1	1,2,3	1	2	-	
CO-2	1,2,3		-	1	
CO-3	1,2,3		1		
CO-4	1,2,3	1			
CO-5	4		2		

Note:
1. Enter correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)
It there is no correlation, put " - "