DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

DBMS LAB MANUAL

Academic year 2024-2025 (odd semester)

Semester –I

Course code **MMC105**

DBMS LAB

Prepared by
Ushasree R

Dr. Manjula Sanjay

HOD-MCA

Dr. Ravishankar M

Principal

Vision of the Institute

To be a centre of excellence in education, research & training and to produce citizens with exceptional leadership qualities to serve national and global needs

Mission of the Institute

To achieve our objectives in an environment that enhances creativity, innovation and scholarly pursuits while adhering to our vision.

Vision of MCA Department

Nurture Continuous Learning through research and innovations in the field of Computer Science, Technology and Applications, to build competent professionals.

Mission of MCA Department

- Create a learning environment to motivate students to build strong technology skills.
- Promote value based ethical practices in all facets of learning
- Instill Entrepreneurial collaborative thinking through structured interventions and industry participation.

Program Education Outcome (PEO's):

PEO1: Analyse real life problems, design computing systems appropriate to its solutions that are technically sound, economically feasible and socially acceptable.
PEO2: Exhibit professionalism, ethical attitude, communication skills, team work in their profession and adapt to current trends by engaging in lifelong learning.
PEO3: Demonstrate Leadership and Entrepreneurship Skills by incorporating organizational goals.

Program Outcome (PO's):

PO1 (Foundation Knowledge): Apply knowledge of mathematics, programming logic and coding fundamentals for solution architecture and problem solving.

PO2 (Problem Analysis): Identify, review, formulate and analyse problems for primarily focussing on customer requirements using critical thinking frameworks.

PO3 (Development of Solutions): Design, develop and investigate problems with as an innovative approach for solutions incorporating ESG/SDG goals.

PO4 (Modern Tool Usage): Select, adapt and apply modern computational tools such as development of algorithms with an understanding of the limitations including human biases.

PO5 (Individual and Teamwork): Function and communicate effectively as an individual or a team leader in diverse and multidisciplinary groups. Use methodologies such as agile.

PO6 (Project Management and Finance): Use the principles of project management such as scheduling, work breakdown structure and be conversant with the principles of Finance for profitable project management.

PO7 (Ethics): Commit to professional ethics in managing software projects with financial aspects. Learn to use new technologies for cyber security and insulate customers from malware.

PO8 (Life-long learning): Change management skills and the ability to learn, keep up with contemporary technologies and ways of working.

Program Specific Outcomes (PSO's):

PSO1: The graduates of the Program will have skills to develop, deploy and maintain applications for desktop, web, mobile, cloud, and cross platforms using modern tools and technologies.

PSO2: The graduates of the program analyze the societal needs to provide novel solutions through technological-based research.

Course Outcomes

CO's	Description	Revised Blooms Level
1	Apply SQL for various database operations	L3
2	Analyze database design with normalization, Concurrency control and transaction management.	L4
3	Design DB schema and Entity Relationship model	L6

Computer Lab Rules and Regulations

DO's

- Come prepared to the Lab.
- Submit your Records to the faculty and sign in the Log Book on entering the Lab
- Observation books have to be brought for all the labs.
- Backlog exercises to be executed after completing regular exercises.
- Regularly attend all the labs
- Put the chairs back to its position before you leave.
- Treat all the devices with care and consideration.
- Behave in a responsible manner at all the times and maintain silence.
- Before leaving the lab shut down the system and rearrange the chairs
- Keep your premises clean

DON'T

- Use Mobile phones and pen drives
- Move around in the lab during the lab session.
- Tamper System Files or Try to access the Server.
- Write Records in the Lab
- Change the system assigned to you without the notice of the Lab Staff.
- Write on the table or mouse pads.
- Do not install or download any software or modify or delete any system files on any lab computers.

Internal evaluation procedure

CIE	Lab	Continuous Evaluation	Observation, Record, Execution Viva	50	Observation -10 Record – 10 Execution of Programs -20 Viva – 10 Total – 50 marks , Scored marks are scaled down to 15 marks	15	7.5	20
		CIE Practical test	Evaluation & Viva-Voce	50	Internal Assessment Tests of 50 Marks, Scored Marks are scaled down to 05 marks	5	2.5	
Total CIE Practical							10	20

COURSE PLAN

List of programs

DBMS: MMC105

SL.NO	Experiments/Programs	Page.No
1	Create the following tables with properly specifying Primary keys, Foreign keys and solve the following queries. BRANCH (Branchid, Branchname, HOD) STUDENT (USN, Name, Address, Branchid, sem) BOOK (Bookid, Bookname, Authorid, Publisher, Branchid) AUTHOR (Authorid, Authorname, Country, age) BORROW (USN, Bookid, Borrowed_Date) **Execute the following Queries:** i.List the details of Students who are all studying in 2nd sem MCA. ii.List the students who are not borrowed any books. iii.Display the USN, Student name, Branch_name, Book_name, Author_name, Books_Borrowed_Date of 2nd sem MCA Students who borrowed books. iv.Display the number of books written by each Author. v.Display the student details who borrowed more than two books. vi.Display the student details who borrowed books of more than one Author. vii.Display the Book names in descending order of their names. viii.List the details of students who borrowed the books which are all published by the same publisher.	
2	Consider the following schema: STUDENT (USN, name, date_of_birth, branch, mark1, mark2, mark3, total, GPA) **Execute the following queries:** i. Update the column total by adding the columns mark1, mark2, mark3. ii. Find the GPA score of all the students. iii. Find the students who born on a particular year of birth from the date_of_birth column. iv. List the students who are studying in a particular branch of study. v. Find the maximum GPA score of the student branch-wise. vi. Find the students whose name starts with the alphabet “S”. vii. Find the students whose name ends with the alphabets “AR”. viii. Delete the student details whose USN is given as 1001.	
3	Design an ER-diagram for the following scenario, Convert the same into a relational model and then solve the following queries. Consider a Cricket Tournament “ABC CUP” organized by an organization. In the	

	tournament there are many teams are contesting each having a Teamid,Team_Name, City, a coach. Each team is uniquely identified by using Teamid. A team can have many Players and a captain. Each player is uniquely identified by Playerid, having a Name, and multiple phone numbers,age. A player represents only one team. There are many Stadiums to conduct matches. Each stadium is identified using Stadiumid, having a stadium_name,Address (involves city,area_name,pincode). A team can play many matches. Each match played between the two teams in the scheduled date and time in the predefined Stadium. Each match is identified uniquely by using Matchid. Each match won by any of the one team that also wants to record in the database. For each match man_of_the match award given to a player. **Execute the following Queries:** i. Display the youngest player (in terms of age) Name, Team name, age in which he belongs of the tournament. ii. List the details of the stadium where the maximum number of matches were played. iii. List the details of the player who is not a captain but got the man_of _match award at least in two matches. iv. Display the Team details who won the maximum matches. v. Display the team name where all its won matches played in the same stadium. Vi. Retrieve the matches played in a specific stadium. Vii. Retrieve all the teams participating in the "ABC CUP" tournament:	
4	Design an ER-diagram for the following scenario, Convert the same into a relational model, normalize Relations into a suitable Normal form and then solve the following queries. A country can have many Tourist places . Each Tourist place is identified by using tourist_place_id, having a name, belongs to a state, Number of kilometers away from the capital city of that state, history. There are many Tourists visits tourist places every year. Each tourist is identified uniquely by using Tourist_id, having a Name, age, Country and multiple emailids. A tourist visits many Tourist places, it is also required to record the visted_date in the database. A tourist can visit a Tourist place many times at different dates. A Tourist place can be visited by many tourists either in the same date or at different dates. **Execute the following Queries:** i. List the state name which is having maximum number of tourist places. ii. List details of Tourist place where maximum number of tourists visited. iii. List the details of tourists visited all tourist places of the state “KARNATAKA”. iv. Display the details of the tourists visited at least one tourist place of the state, but visited all states tourist places. v. Display the details of the tourist place visited by the tourists of all country.	
5	A country wants to conduct an election for the parliament. A country having many constituencies. Each constituency is identified uniquely by Constituency_id, having the Name, belongs to a state,Number_of_voters. A constituency can have many voters. Each voter is uniquely identified by using Voter_id, having the Name, age, address (involves Houseno,city,state,pincode). Each voter belongs to only one constituency. There are many candidates contesting in the election. Each candidates	

	are uniquely identified by using candidate_id, having Name, phone_no, age, state. A candidate belongs to only one party.Thereare many parties. Each party is uniquely identified by using Party_id, having Party_Name,Party_symbol. A candidate can contest from many constituencies under a same party. A party can have many candidates contesting from different constituencies. No constituency having the candidates from the same party. A constituency can have many contesting candidates belongs to different parties. Each voter votes only one candidate of his/her constituencty. **Execute the following Queries:** i. List the details of the candidates who are contesting from more than one constituencies which are belongs to different states. ii. Display the state name having maximum number of constituencies. iii. Create a stored procedure to insert the tuple into the voter table by checking the voter age. If voter's age is at least 18 years old, then insert the tuple into the voter else display the "Not an eligible voter msg" . iv. Create a stored procedure to display the number_of_voters in the specified constituency. Where the constituency name is passed as an argument to the stored procedure. v. Create a TRIGGER to UPDATE the count of " Number_of_voters" of the respective constituency in "CONSTITUENCY" table , AFTER inserting a tuple into the "VOTERS" table.	
6	**Execute the fallowing schema using MongoDB.** Exercise–I: Student Database Agenda: Create database, create collection, insert data, find, find one, sort, limit, skip, distinct, projection. Create a student database with the fields: (SRN, Sname, Degree, Sem, CGPA) 1.Display all the documents 2.Display all the students in MCA 3.Display all the students in ascending order 4.Display first 6 students 5.Display students 5,6,7 6.list the degree of student "Ram" 7.Display student's details of 4,5,6,7 in descending order of percentage 8.Display the number of students in MCA 9.Display all the degrees without _id 10.Display all the distinct degrees	

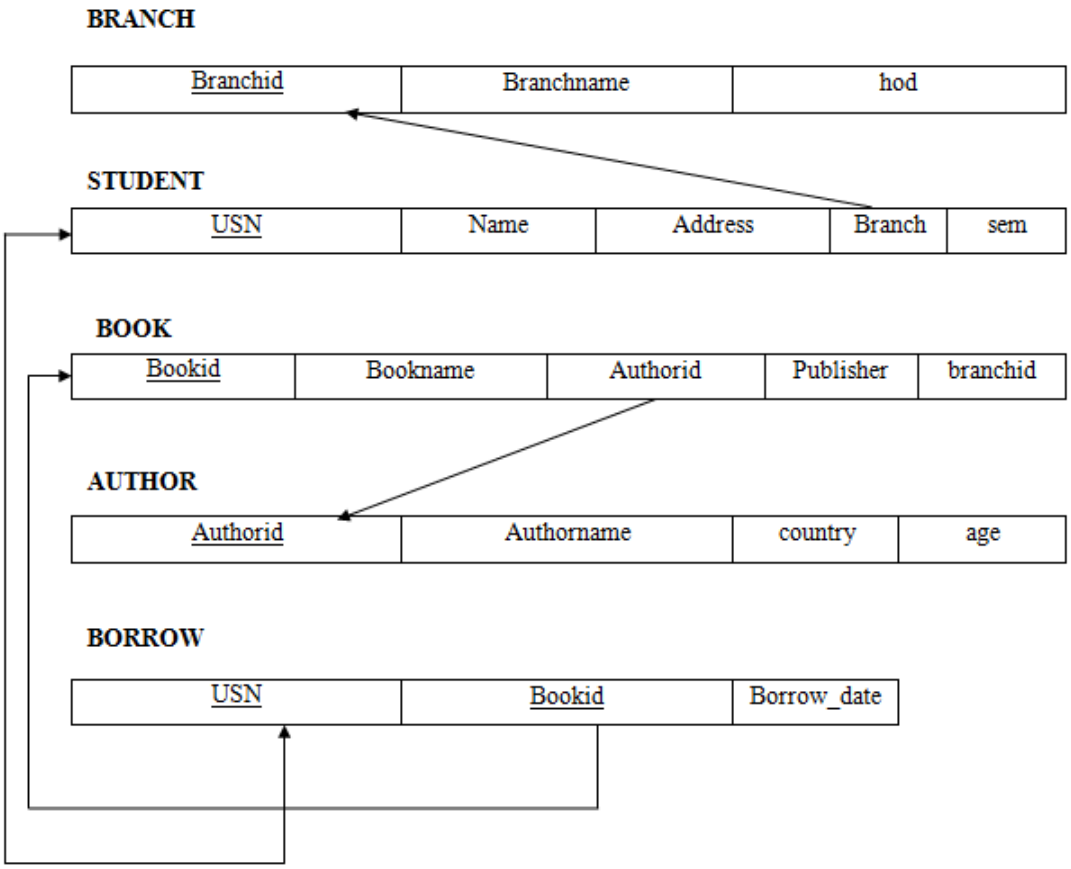
1. Create the following tables with properly specifying primary keys, Foreign keys and solve Following queries:

BRANCH (Branchid,Branchname,Hod)
STUDENT (USN,Name,Address,Branchid,sem)
Author (Authorid,Authorname,Country,Age)
Book(Bookid,Book_name,Authorid,Publisher,Branchid)
BORROW (USN,Boookid, Borrowed_Date)

Execute the following queries:

i) List the details of student who are all studying in 2nd sem MCA.
ii) List the students who are not borrowed any books
iii) Display the USN, Student name, Branch_name, book_name, author_name, books_ borrowed _date of 2nd sem MCA students who borrowed books.
iv) Display the number of books written by each Author.
v) Display the student details who borrowed more than two books.
vi) Display the student details who borrowed books of more than one Author.
vii) Display the Book names in descending order of their names.
viii) List the details of students who borrowed the books which are all published by the Same Publisher.

SCHEMA DISGRAM



SOLUTIONS

```
create table branch
(branchid int primary key,
bname varchar(10),
hod varchar(10));

create table student
(usn varchar(10) primary key,
name varchar(10),
addr varchar(15),
branchid int references branch(branchid),
sem int);

create table book
(bookid int primary key,
bname varchar(10),
 Authorid number(4) references author(authorid),
Publisher varchar2(10),
branchid number(4) references branch(branchid));


create table author
```

```
(authorid int primary key,
aname varchar(10),
country varchar(10),
age int);

create table borrow
(usn varchar(10) references student(usn),
bookid int references book(bookid),
borrowdate date);
```

```
SQL> select * from branch;
BRANCHID        BNAME       HOD
----------      ----------  ----------
1               mca         Rama
2               mba         Ravi
3               cse         Praveen
```

```
SQL> select * from student;
USN       NAME        ADDR        BRANCHID        SEM
----------  ---------- ----------          ----------      ----------
1rn1      harish      bangalore           1               2
1rn2       bharath     mysore              2               3
1rn3       kiran       delhi               3               6
1rn4       mahi        chennai             4               7
```

```
SQL> select * from book;

BOOKID          BNAME       AUTHORID    PUBLISHER       BRANCHID
---------- ---------- ---------- ---------- ----------
1111            c prog      123         pearson         1
2222             dbms       124         mgrawhill       2
3333            oops 1      25          sapna           3
4444            unix        126          subhash         4
5555            cprog        127         pearson         5
```

```
SQL> select * from author;
AUTHORID        ANAME       COUNTRY     AGE
---------- ---------- ---------- ----------
123             navathe       india       55
124              ritche       uk          44
125             RAM          india       55
```

```
SQL> select * from borrow;
USN       BOOKID      BORROWDATE
```

---------- ---------- ---------
1rn1 2222 10-JAN-00
1rn1 3333 05-MAR-16
1rn3 5555 01-JUN-10
1rn5 2222 19-MAY-00
1rn2 1111 22-FEB-15

Query 1.

select * from student where sem=2 and branchid in(select branchid from branch where bname='mca')

USN	NAME	ADDR	BRANCHID	SEM
1rn1	harish	bangalore	1	2

Query 2.

select * from student where usn not in (select usn from borrow);

USN	NAME	ADDR	BRANCHID	SEM
1rn4	mahi	Chennai	4	7

Query 3.

select student.usn, student.name,branch.bname, book.bname, borrow borrowdate from student , branch, book, author, borrow where student.usn=borrow.usn and borrow.bookid=book.bookid and book.authorid= author.authorid and student.branchid=branch.branchid and student.sem=2 and branch.bname='mca';

USN	NAME	BNAM	BNAME	ANAME	BORROWDAT
1rn1	harish	mca	dbms	ritche	10-JAN-20
1rn1	harish	mca	oops	Ravi	05-MAR-16

Query 4

select count(*) , authorid from book group by authorid;

COUNT(*)	AUTHORID
1	123
1	125
1	124

Query 5

select * from student where usn in (select usn from borrow group by usn having count(usn) >=2);

USN	NAME	ADDR	BRANCHID	SEM
1rn1	harish	bangalore	1	2

Query 6.

select * from student s where exists (select br.usn from borrow br join book bk on br.bookid=bk.bookid where br.usn=s.usn group by usn having count(distinct authorid)>1);

USN	NAME	ADDR	BRANCHID	SEM
1rn1	harish	bangalore	1	2

Query 7.

select bname from book order by bname desc;

BNAME
unix
oops
dbms
cprog
c prog

Query 8.

select * from student s where exists (select usn , publisher from borrow join book on borrow.bookid=book.bookid where s.usn=borrow.usn group by usn having count(distinct publisher)=1);

USN	NAME	ADDR	BRANCHID	SEM
1rn2	bharath	mysore	2	3
1rn3	kiran	delhi	3	6
1rn5	krishna	hubli	5	4

2. Consider the following schema:

STUDENT (USN, name, date_of_birth, branch, mark1, mark2, mark3, total, GPA)

Execute the following queries:

1. Update the column total by adding the columns mark1, mark2, mark3.
2. Find the GPA score of all the students.
3. Find the students who born on a particular year of birth from the date_of_birth column.
4. List the students who are studying in a particular branch of study.
5. Find the maximum GPA score of the student branch-wise.
6. Find the students whose name starts with the alphabet “S”.
7. Find the students whose name ends with the alphabets “AR”.
8. Delete the student details whose USN is given as 1001.

SOLUTIONS

i. Update the column total by adding the columns mark1, mark2, mark3.
UPDATE STUDENT SET total = mark1 + mark2 + mark3;

ii. Find the GPA score of all the students.
UPDATE STUDENT SET GPA = total / 300 * 100;

iii. Find the students who born on a particular year of birth from the date_of_birth column.
SELECT * FROM STUDENT WHERE strftime('%Y', date_of_birth) = '1998'; (OR)
SELECT date_of_birth from STUDENT where substr(student.dob,-2)=’99’;

iv. List the students who are studying in a particular branch of study.
SELECT * FROM STUDENT WHERE branch = 'MCA';

v. Find the maximum GPA score of the student branch-wise.
SELECT s.USN, s.branch, s.GPA FROM STUDENT s WHERE (s.branch, s.GPA) IN (
SELECT branch, MAX(GPA) AS max_gpa FROM STUDENT GROUP BY branch);

vi. Find the students whose name starts with the alphabet “S”.
SELECT * FROM STUDENT WHERE name LIKE 'S%';

vii. Find the students whose name ends with the alphabets “AR”.
SELECT * FROM STUDENT WHERE name LIKE '%AR';

viii. Delete the student details whose USN is given as 1001.
DELETE FROM STUDENT WHERE USN = '1001';

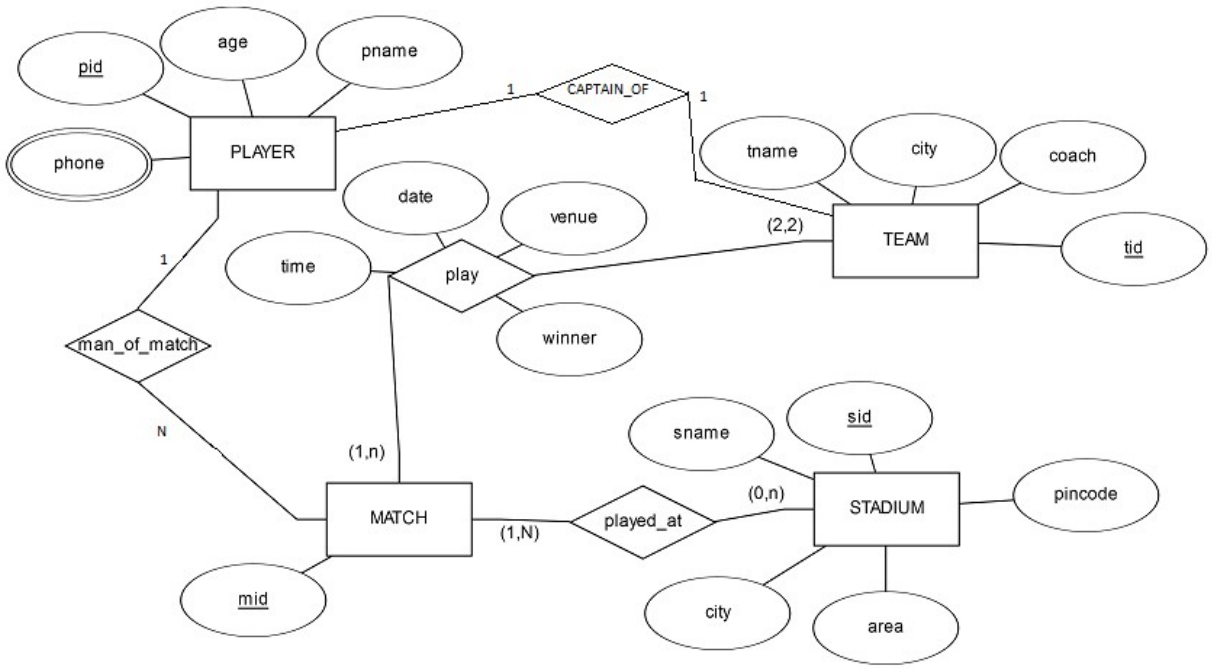
3. Design an ER-diagram for the following scenario, Convert the same into a relational model and then solve the following queries.

Consider a Cricket Tournament “ABC CUP” organized by an organization. In the tournament there are many teams are contesting each having a Teamid,Team_Name, City, a coach. Each team is uniquely identified by using Teamid. A team can have many Players and a captain. Each player is uniquely identified by Playerid, having a Name, and multiple phone numbers,age. A

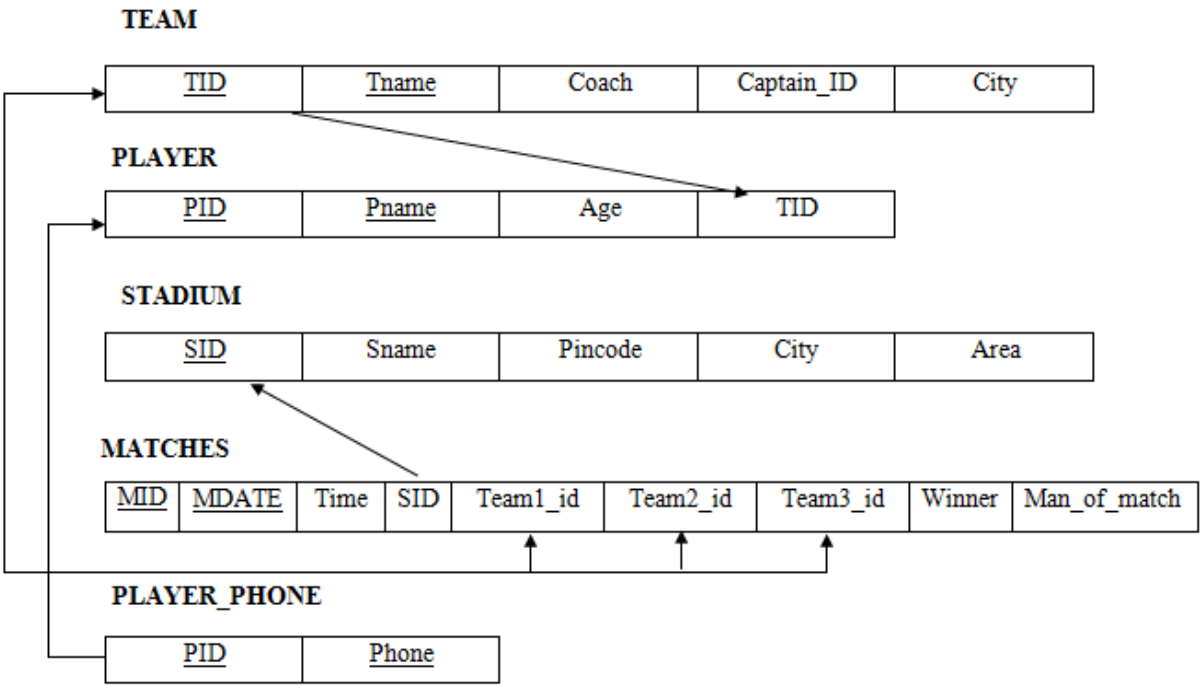
player represents only one team. There are many Stadiums to conduct matches. Each stadium is identified using Stadiumid, having a stadium_name,Address (involves city,area_name,pincode). A team can play many matches. Each match played between the two teams in the scheduled date and time in the predefined Stadium. Each match is identified uniquely by using Matchid. Each match won by any of the one team that also wants to record in the database. For each match man_of_the match award given to a player. Execute the following Queries:

i. Display the youngest player (in terms of age) Name, Team name, age in which he belongs of the tournament.
ii. List the details of the stadium where the maximum number of matches were played.
iii. List the details of the player who is not a captain but got the man_of _match award at least in two matches.
iv. Display the Team details who won the maximum matches.
v. Display the team name where all its won matches played in the same stadium.
Vi. Retrieve the matches played in a specific stadium.
Vii. Retrieve all the teams participating in the "ABC CUP" tournament:

ER DIAGRAM



SCHEMA DISGRAM



SOLUTIONS

```
create table team
( tid int primary key,
tname varchar(20),
coach varchar(20),
captain_pid int,
city varchar(20));

create table player
( pid int primary key,
pname varchar(2),
age int,
tid int references team(tid))

create table stadium
(sid int primary key,
sname varchar(20),
picode number(8),
city varchar(20),
area varchar(20));
```

```
create table match
(mid int primary key,
mdate date,
time varchar(6),
sid int references stadium(sid),
team1_id int references team(tid),
team2_id int references team(tid),
winning_team_id int references team(tid), man_of_match int references player(pid));


create table player_phone
( pid int references player(pid),
phone int ,
primary key(pid,phone));
```

SQL> select * from team;

TID	TNAME	COACH	CAPTAIN_PID	CITY
123	rcb	sunil	1	bangalore
12	csk	laxman	3	chennai
125	royals	singh	4	rajasthan
126	daredevils	sehwag	2	delhi

SQL> select * from player;

PID	PNAME	AGE	TID
1	sachin	33	123
2	dravid	32	124
3	dhoni	30	124
4	raina	30	125
5	kohli	23	126

SQL> select * from stadium;

SID	SNAME	PICODE	CITY	AREA
111	chinnaswamy	56001	bangalore	mg road
222	kotla	460009	delhi	highway
333	international	38883	chennai	tr nagar
444	ksca	560098	bangalore	peenya
555	csca	567772	cochin	beach road

SQL> select * from match;

MID	MDATE	TIME	SID	TEAM1_ID	TEAM2_ID	TEAM3_ID	WIN	M_of the match
100	10-JAN-17	10am	111	123	124	120	1	1
102	11-JAN-17	11am	222	124	126	122	2	5

SQL> select * from player_phone;

PID	PHONE
1	998882928
2	877563733
2	988928822
3	877366383

Query 1

Select pname, tname, age from player p, team t where p.tid=t.tid and age = (select min(age) from player);

PNAME	TNAME	AGE
kohli	daredevils	23

Query 2:

select * from stadium where sid in (select sid from match group by sid having count(sid) = (select max(count(sid)) from match group by sid))

SID	SNAME	PICODE	CITY	AREA
111	chinnaswamy	56001	bangalore	mg road

Query 3:

select * from player where pid not in (select captain_pid from team) and pid in (select man_of_match from match group by man_of_match having count(man_of_match)=2);

PID	PNAME	AGE	TID
5	kohli	23	126

Query 4:

select * from team where tid in (select winning_team_id from match group by winning_team_id having count(winning_team_id)=(select max(count(winning_team_id)) from match group by winning_team_id))

TID	TNAME	COACH	CAPTAIN	PID	CITY
126	daredevils	sehwag	Sachin	2	delhi

Query 5:

select tname from team where tid in (select winning_team_id from match group by(winning_team_id,sid) having count(*) in (select count(winning_team_id) from match group by winning_team_id))

TNAME
Rcb

Query 6:

Vi. Retrieve the matches played in a specific stadium.

Select * from match where statiumid=1;

Query 7:

vii. Retrieve all the teams participating in the "ABC CUP" tournament:

select * from team where teamid in (select distinct team1_id from match union select distinct team2_id from match);

4. Design an ER-diagram for the following scenario, Convert the same into a relational model, normalize Relations into a suitable Normal form and then solve the following queries. A country can have many Tourist places . Each Tourist place is identified by using tourist_place_id, having a name, belongs to a state, Number of kilometers away from the capital city of that state, history. There are many Tourists visits tourist places every year. Each tourist is identified uniquely by using Tourist_id, having a Name, age, Country and multiple emailids. A tourist visits many Tourist places, it is also required to record the visted_date in the database. A tourist can visit a Tourist place many times at different dates. A Tourist place can be visited by many tourists either in the same date or at different dates.

Queries:

i. List the state name which is having maximum number of tourist places.

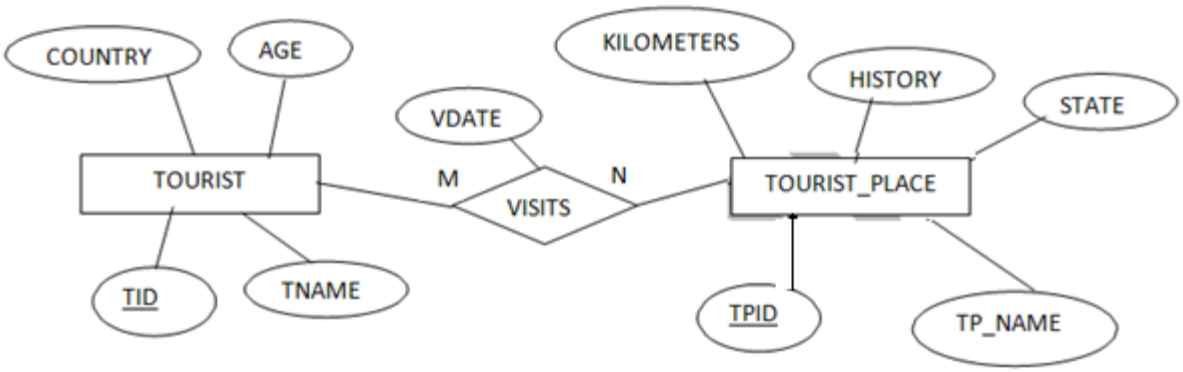
ii. List details of Tourist place where maximum number of tourists visited.

iii. List the details of tourists visited all tourist places of the state “KARNATAKA”.

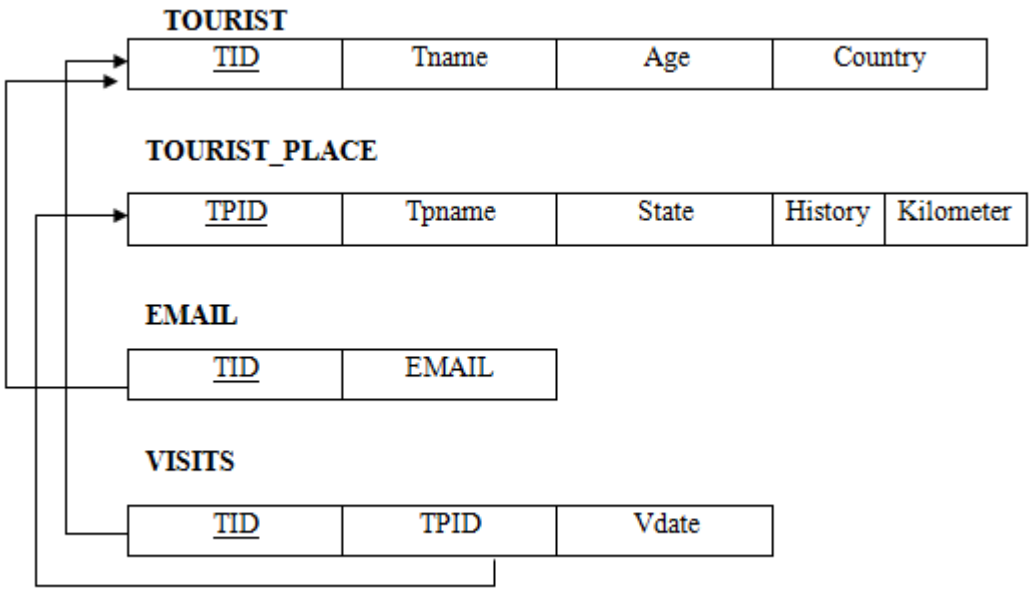
iv. Display the details of the tourists visited at least one tourist place of the state, but visited all states tourist places.

v. Display the details of the tourist place visited by the tourists of all country.

ER DIAGRAM



SCHEMA DISGRAM



```
create table tourist_place
(tpid number primary key,
history varchar(20),
kilometers number(3)
,state varchar(20),
tpname varchar(20));

create table tourist(tid number primary key,
 country varchar(20), age number, tname varchar(20));
create table visits
(tpid number(3) references tourist_place(tpid),tid number references tourist(tid), vdate date,
     primary key(tpid,tid));

create table email (tid number references tourist(tid), email varchar(20),primary key(tid,email));


SQL> insert into tourist (tid,country,age,tname) values('22','india','34','prakash');
     1 row created.
     Note: Insert 5 records

     SQL> insert into tourist_place values('22','india','34','prakash');
     1 row created.




     SQL> select * from tourist;
     TID COUNTRY AGE TNAME
     ---------- -------------------- ---------- --------------------
     22 india 34 prakash
     23 orissa 28 bhanu
     24 india 30 nagesh

     SQL> insert into visits values('&tpid','&tid','&vdate');
     Enter value for tpid: 12
     Enter value for tid: 23
     Enter value for vdate: 13-nov-2014
     old 1: insert into visits values('&tpid','&tid','&vdate')
     new 1: insert into visits values('12','23','13-nov-2014')
     1 row created.

     SQL> select * from visits;
     TPID TID VDATE
     ---- ---------- ---------
     12 23 13-NOV-14
     11 24 24-JUN-13
```

13 22 25-SEP-11
11 23 23-FEB-10
13 23 12-JAN-10
14 24 10-JAN-17

SQL> insert into email values('&tid','&email');
Enter value for tid: 23
Enter value for email: bhanu12@gmail.com
old 1: insert into email values('&tid','&email')
new 1: insert into email values('23','bhanu12@gmail.com')
1 row created.

SQL> select * from email;
TID EMAIL
---------- --------------------
23 bhanu12@gmail.com
22 prakash242@gmail.com
24 nageshh@gmail.com

Query 1:
select state from tourist_place group by state having count(state)=(select max(count(state)) from tourist_place group by state);

STATE

Karnataka

query 2:

select * from tourist_place where tpid in (select tpid from visits group by tpid having count(tpid)= (select max(count(tpid)) from visits group by tpid));

TPID	HISTORY	KILOMETERS	STATE	TPNAME
11	beauty	160	karnataka	ooty
13	beach	360	tamilnadu	marina

query 3:
select * from tourist t where t.tid in (select tid from visits join tourist_place on visits.tpid=tourist_place.tpid where state='karnataka' group by tid having count(state) in (select count(state) from tourist_place where state='karnataka'));

TID	COUNTRY	AGE	TNAME
24	india	30	nagesh

query 4:

select * from tourist t where t.tid in (select tid from visits join tourist_place on visits.tpid=tourist_place.tpid group by tid having count(distinct state) in (select count(distinct state) from tourist_place));

TID	COUNTRY	AGE	TNAME
23	orissa	28	bhanu

query 5:

select * from tourist_place where tpid in(select tpid from visits join tourist on visits.tid=tourist.tid group by tpid having count(distinct country)= (select count(distinct country) from tourist));

TPID	HISTORY	KILOMETERS	STATE	TPNAME
11	beauty	160	karnataka	ooty
13	beach	360	tamilnadu	marina

5. A country wants to conduct an election for the parliament. A country having many constituencies. Each constituency is identified uniquely by Constituency_id, having the Name, belongs to a state,Number_of_voters. A constituency can have many voters. Each voter is uniquely identified by using Voter_id, having the Name, age, address (involves Houseno,city,state,pincode). Each voter belongs to only one constituency. There are many candidates contesting in the election. Each candidates are uniquely identified by using candidate_id, having Name, phone_no, age, state. A candidate belongs to only one party.Thereare many parties. Each party is uniquely identified by using Party_id, having Party_Name,Party_symbol. A candidate can contest from many constituencies under a same party. A party can have many candidates contesting from different constituencies. No constituency having the candidates from the same party. A constituency can have many contesting candidates belongs to different parties. Each voter votes only one candidate of his/her constituencty.

Execute the following queries:

i. List the details of the candidates who are contesting from more than one constituencies which are belongs to different states.

ii. Display the state name having maximum number of constituencies.

iii. Create a stored procedure to insert the tuple into the voter table by checking the voter age. If voter's age is at least 18 years old, then insert the tuple into the voter else display the "Not an eligible voter msg" .

iv. Create a stored procedure to display the number_of_voters in the specified constituency. Where the constituency name is passed as an argument to the stored procedure.

v. Create a TRIGGER to UPDATE the count of " Number_of_voters" of the respective constituency in "CONSTITUENCY" table , AFTER inserting a tuple into the "VOTERS" table.

SOLUTIONS

```
create table constituency (cons_id number(20) primary key, csname varchar(20),
csstate varchar(20), no_of_voters number(10));


create table party
(pid number(20) primary key, pname varchar(20),
psymbol varchar(10));

create table candidates (cand_id number(12) primary key, phone_no number(10),
age number(2), state varchar(20), name varchar(20),
pid int references party(pid));


create table contest
(cons_id number(20) references constituency(cons_id), cand_id number(12) references
candidates(cand_id) primary key(cons_id,cand_id);

create table voter
(vid number(20) primary key, vname varchar(20), vage number(5), vaddr varchar(20),
cons_id number(20) references constituency(cons_id), cand_id number(12) references
candidates(cand_id));


SQL > select * from constituency;
SQL > select * from party;
SQL > select * from candidates;
SQL > select * from contest;
SQL > select * from voters;
```

Query 1:

```
Select * from candidates where cand_id in (select * from contest join constituency on
contest.cons_id=constituency.cons_id)
```

Query 2:

```
select csstate from constituency group by csstate having count(csstate) in (select
max(count(csstate)) from constituency group by csstate);
```

Query 3:

```
create or replace procedure agechecking ( id in number,age in number)
as BEGIN
if age>18 then
insert into voter(vid,vage) values (id,age);
```

```
 else
dbms_output.put_line('age should be high');
end if;
end  agechecking;

/
```
Procedure created.

SQL> set server output on; SQL> exec age checking (25,21);

PL/SQL procedure successfully completed. // row inserted

SQL> exec age checking (20,15);
age should be high //Message displayed as age is less than or equal to 18
PL/SQL procedure successfully completed.

query 4:

```
create or replace procedure display_count ( const_id number)
as
vid constituency.cons_id % type;
begin
select no_of_voters into vid from constituency where cons_id = const_id and rownum = 1;
dbms_output.put_line ( 'total voters are: ' || vid);
end;
/
```
Procedure created.

SQL> select * from constituency;
SQL> exec display_count(111);
Total voters are: 2

Query 5:

```
create or replace trigger count after insert on voter
for each row begin
update constituency set no_of_voters = no_of_voters + 1 where cons_id=:new.cons_id;
end count;
/
```

Trigger created.

```
SQL> set serveroutput on;
SQL> select * from constituency;
SQL> insert into voter values(348,'nagesh',30,'mandya',111,121);
1 row created.
```

After insertion into voter table, the constituency table isautomatically updated.

SQL> select * from constituency;

6. Execute the fallowing schema using MongoDB.

Exercise–I:

Student Database Agenda: Create database, create collection, insert data, find, find one, sort, limit, skip, distinct, projection.

Create a student database with the fields: (SRN, Sname, Degree, Sem, CGPA)

1.Display all the documents
2.Display all the students in MCA
3.Display all the students in ascending order
4.Display first 6 students
5.Display students 5,6,7
6.list the degree of student "Ram"
7.Display student's details of 4,5,6,7 in descending order of percentage
8.Display the number of students in MCA
9.Display all the degrees without _id
10.Display all the distinct degrees

Exercise: Student Database

Agenda : Create database ,create collection ,insert data ,find ,find one ,sort, limit , skip, distinct, projection

```
>use student1
```

Switched to db student1

```
>db.studcol1.insert({srn:110,sname:””Rahul”,degree:”MCA”,sem:6,CGPA:7.9})
```

OR

```
>doc1=({srn:110,sname:”rahul,degree:”MCA”,sem:6,CGPA:7.9})
>db.studcol1.insert(doc1);
```

NOTE: Insert 10 Documents.

SOLUTIONS

1. Display all the document
   >db.studcol1.find()

2. Display all the students in MCA
   >db.studcol1.find({degree:”MCA”})

3. Display all the students in ascending order

```
>db.studcol1.find().sort({sname:1})
```

4. Display first 5 students

```
>db.studcol1.find().limit(5)
```

5.display students 5,6,7

```
>db.studcol1.find().skip(4).limit(3)
```

6.List the degree of student “Rahul”

```
>db.studcol1.find({degree:1,sname:”Rahul”})
```

7.Display students of 5,6,7 in descending order of percentage.

```
>db.studcol1.find().sort({CGPA:1}).skip(4).limit(3)
```

8.Display the number of students in MCA

```
>db.studcol1.find({degree:”MCA”}).count()
```

9.Display all the degrees without _id

```
>db.studcol1.find({},{_id:0})
```

10.Display all the distinct degrees

```
>db.studcol1.distinct(“degree”)
```

DBMS VIVA QUESTIONS

1. What is database?

A database is a logically coherent collection of data with some inherent meaning, representing some aspect of real world and which is designed, built and populated with data for a specific purpose.

2. What is DBMS?

It is a collection of programs that enables user to create and maintain a database. In other words it is general-purpose software that provides the users with the processes of defining, constructing and manipulating the database for various applications.

3. Disadvantage in File Processing System?

- Data redundancy & inconsistency.
- Difficult in accessing data.
- Data isolation.
- Data integrity.
- Concurrent access is not possible.

- Security Problems.

4. **Advantages of DBMS?**

- Providing multiple user interfaces.
- Enforcing integrity constraints.
- Providing backup and recovery.
- Redundancy is controlled.
- Unauthorized access is restricted

5. Describe the three levels of data abstraction?

There are three levels of abstraction:

- Physical level: The lowest level of abstraction describes how data are stored.
- Logical level*:* The next higher level of abstraction, describes what data are stored in database and what relationship among those data.
- View level: The highest level of abstraction describes only part of entire database.

6. Define the "integrity rules"

There are two Integrity rules.

- Entity Integrity: States that "Primary key cannot have NULL value"
- Referential Integrity: States that "Foreign Key can be either a NULL value or should be Primary Key value of other relation.

7. What is Data Independence?

Data independence means that "the application is independent of the storage structure and access strategy of data". In other words, the ability to modify the schema definition in one level should not affect the schema definition in the next higher level.

Two types of Data Independence:

- Physical Data Independence: Modification in physical level should not affect the logical level.
- Logical Data Independence: Modification in logical level should affect the view level.

8. What is a view? How it is related to data independence?

- A view may be thought of as a virtual table, that is, a table that does not really exist in its own right but is instead derived from one or more underlying base table. In other words, there is no stored file that direct represents the view instead a definition of view is stored in data dictionary.
- Growth and restructuring of base tables is not reflected in views. Thus the view can insulate users from the effects of restructuring and growth in the database. Hence accounts for logical data independence.

9. What is data model?

- This data model is based on real world that consists of basic objects called entities and of relationship among these objects. Entities are described in a database by a set of attributes.

10. What is an Entity?

- It is an 'object' in the real world with an independent existence.

11. What is an Entity type?

- It is a collection (set) of entities that have same attributes.

12. What is DML (Data Manipulation Language)?

- DML is a language that enables users to access or manipulate data as organized by the appropriate data model.
- Procedural DML or Low level: DML requires a user to specify what data are needed and how to get those data.
- Non-Procedural DML or High level: DML requires a user to specify what data are needed without specifying how to get those data.

13. What is normalization?

It is a process of analyzing the given relation schemas based on their Functional Dependencies (FDs) and primary key to achieve the properties

- Minimizing redundancy
- Minimizing insertion, deletion and update anomalies.

14. What is Functional Dependency?

A Functional dependency is denoted by X Y between two sets of attributes X and Y that are subsets of R specifies a constraint on the possible tuple that can form a relation state r of R. The constraint is for any two tuples t1 and t2 in r if t1[X] = t2[X] then they have t1[Y] = t2[Y]. This means the value of X component of a tuple uniquely determines the value of component Y.

15. When is a functional dependency F said to be minimal?

Every dependency in F has a single attribute for its right hand side.

- We cannot replace any dependency X A in F with a dependency Y A where Y is a proper subset of X and still have a set of dependency that is equivalent to F.
- We cannot remove any dependency from F and still have set of dependency that is equivalent to F.

16. What is Lossless join property?

- It guarantees that the spurious tuple generation does not occur with respect to relation schemas after decomposition.

17. What is 1 NF (Normal Form)?

- The domain of attribute must include only atomic (simple, indivisible) values.

18. What is Fully Functional dependency?

- It is based on concept of full functional dependency. A functional dependency X Y is fully functional dependency if removal of any attribute A from X means that the dependency does not hold any more.

19. What is 2NF?

- A relation schema R is in 2NF if it is in 1NF and every non-prime attribute A in R is fully functionally dependent on primary key.

20. What is 3NF?

- A relation schema R is in 3NF if it is in 2NF and for every FD X A either of the following is true In other words, if every non prime attribute is non-transitively dependent on primary key.
  - X is a Super-key of R.
  - A is a prime attribute of R.

21. What is BCNF (Boyce-Codd Normal Form)?

- A relation schema R is in BCNF if it is in 3NF and satisfies additional constraints that for every FD X A, X must be a candidate key.

22. What is trigger?

- Database triggers are sets of commands that get executed when an event(Before insert , occurs on a table

23. Difference between TRUNCATE, DELETE and DROP commands

- The DELETE command is used to remove rows from a table. A WHERE clause can be used to only remove some rows. If no WHERE condition is specified, all rows will be removed.
- TRUNCATE removes all rows from a table. The operation cannot be rolled back and no triggers will be fired. As such, TRUCATE is faster and doesn't use as much undo space as a DELETE.
- The DROP command removes a table from the database. All the tables' rows, indexes and privileges will also be removed. No DML triggers will be fired. The operation cannot be rolled back.
- DROP and TRUNCATE are DDL commands, where as DELETE is a DML command.
- Therefore DELETE operations can be rolled back (undone), while DROP and TRUNCATE operations cannot be rollback.

24. Mention the different languages present in DBMS

The different languages present in DBMS are as follows:

- **DDL (Data Definition Language)** – Consists of commands which are used to define the database.
- **DML (Data Manipulation Language)** – Consists of commands which are used to manipulate the data present in the database.
- **DCL (Data Control Language)** – Consists of commands which deal with the user permissions and controls of the database system.

- **TCL (Transaction Control Language)** – Consist of commands which deal with the transaction of the database.

25. What do you understand by the terms Entity, Entity Type, and Entity Set in DBMS?

- **Entity:** An entity is a real-world object having attributes, which are nothing but characteristics of that particular object. For example, an employee can be an entity. This particular entity can have attributes such as empid, empname, etc.
- **Entity Type:** Entity type is nothing but a collection of entities, having the same attributes. Generally, an entity type refers to one or more related tables in a particular database. So, you can understand entity type as a characteristic which uniquely identifies the entity. For example, an employee can have attributes such as empid, empname, department, etc.
- **Entity Set:** An entity set is the collection of all the entities of a particular entity type in a database. For example, a set of employees, a set of companies, and a set of people can come under an entity set.

25. What are the different types of keys in the database?

- **Primary Key –** This is a set of attributes which are used to uniquely identify every tuple.
- **Foreign Key –** An attribute that can only take the values present as the values of some other attribute, is the foreign key to the attribute to which it refers.
- **Composite Key –** A composite key is a combination of two or more columns that identify each tuple uniquely.
- **Candidate Key –** This is a set of attributes which can uniquely identify a table. Each table can have more than a candidate key. Apart from this, out of all the candidate keys, one key can be chosen as the Primary key
- **Super Key –** This is a set of attributes which can uniquely identify a tuple. So, a candidate key, primary key, and a unique key is a super key, but vice-versa isn't true.

25. What are the stored procedure

- A group of SQL statements which can be reused again and again. These statements are created and stored in the database. We can execute stored procedures by using the exec command, whenever we want.

26. What is E-R model in the DBMS?

- E-R model is known as an Entity-Relationship model in the DBMS which is based on the concept of the Entities and the relationship that exists among these entities.

27. What are different types of joins in SQL?

There are 4 types of SQL Joins:

- **Inner Join:** This type of join is used to fetch the data among the tables which are common in both the tables.

- **Left Join:** This returns all the rows from the table which is on the left side of the join but only the matching rows from the table which is on the right side of the join.
- **Right Join:** This returns all the rows from the table which is on the right side of the join but only the matching rows from the table which is on the left side of the join.
- **Full Join:** This returns the rows from all the tables on which the join condition has put and the rows which do not match hold null values.

28. What is MongoDB?

- MongoDB is an open-source document-oriented NoSQL database that was created in 2007 by Dwight Merriman, Eliot Horowitz, and Kevin Ryan. Rather than tables, columns, and rows, MongoDB is based on collections and JSON-like documents instead of databases.

29. What are MongoDB Documents and Collections?

- A document in the MongoDB is an object that represents a single record and it is an analogous to a row in a SQL table. MongoDB document has a key/value structure
- A Collection is a set of related documents; it acts as the equivalent of RDBs (relational database) tables. MongoDB database is simply a group of collections that hold a set of similar or partially similar documents

30. **What is a document in MongoDB?**

- In MongoDB, a document is a data structure composed of field-and-value pairs. Documents are akin to rows in relational databases, but they can contain nested structures, arrays, and other complex data types.

31. What is a CRUD operation in MongoDB?

- A CRUD operation refers to the basic operations performed on data in a database: Create Read, Update, and Delete.

32. What is the query language used in MongoDB?

- MongoDB uses a query language called MongoDB Query Language (MQL). MQL is based on JSON-like syntax and provides a rich set of query operators for data manipulation and retrieval.

33. What are the data types in MongoDB?

MongoDB supports a wide range of data types as values in documents. Documents in MongoDB are similar to objects in JavaScript. Along with JSON's essential key/value–pair nature, MongoDB adds support for a number of additional data types. The common data types in MongoDB are:

- Null
  {"x" : null}
- Boolean
  {"x" : true}
- Number
  {"x" : 4}
- String
  {"x" : "foobar"}

34. What is field is a database?

A field is an area within a record reserved for a specific piece of data. Example: Employee Name, Employee ID etc.

35. What are the advantages on stored procedure?

- Stored procedures are precompiled and stored in the database. This enables the database to execute the queries much faster ,since many queries can be included in a stored procedure ,round trip time to execute multiple queries from source code to database and back is avoided.