## **ADBMS Practical 2**

# Using practical 1

1. Count the customers with grades above New York's average.

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

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

```
mysql> delete from orders where salesman_id = 5006;
Query OK, 1 row affected (0.00 sec)
mysql> delete from customer where salesman_id = 5006;
Query OK, 1 row affected (0.01 sec)
mysql> delete from salesman where salesman_id = 5006;
Query OK, 1 row affected (0.00 sec)
```

2. Design ERD for the following schema and execute the following Queries on it

#### Actor

```
mysql> CREATE TABLE ACTOR (
-> ACT_ID INT (3) PRIMARY KEY,
-> ACT_NAME VARCHAR (20),
-> ACT_GENDER CHAR (1)
-> )
```



mysql> select *from actor;					
ACT_ID	ACT_NAME	ACT_GENDER			
302   303	ANUSHKA PRABHAS PUNITH JERMY	į F			
+	set (0.00	+			

### **Director**

mysql> CREATE TABLE DIRECTOR (

- -> DIR\_ID INT (3) PRIMARY KEY,
- -> DIR\_NAME VARCHAR (20),
- -> DIR\_PHONE INT (10)
- ->);



```
mysql> select *from director;

| DIR_ID | DIR_NAME | dir_phone |

| 60 | RAJAMOULI | 875161100 |

| 61 | HITCHCOCK | 776613891 |

| 62 | FARAN | 998677653 |

| 63 | STEVEN SPIELBERG | 898977653 |

4 rows in set (0.00 sec)
```

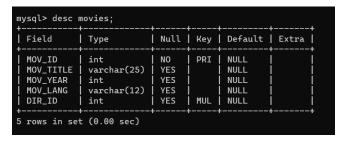
#### **Movies**

mysql> CREATE TABLE MOVIES (

- -> MOV\_ID INT (4) PRIMARY KEY,
- -> MOV\_TITLE VARCHAR (25),
- -> MOV\_YEAR INT (4),
- -> MOV\_LANG VARCHAR (12),
- -> DIR\_ID INT (3),

-> FOREIGN KEY (DIR\_ID) REFERENCES DIRECTOR (DIR\_ID)

->);



mysql> select *from movies;						
MOV_ID	MOV_TITLE	MOV_YEAR	MOV_LANG	DIR_ID		
1001   1002   1003   1004	BAHUBALI-2   BAHUBALI-1   AKASH   WAR HORSE	2008	TELAGU TELAGU KANNADA ENGLISH	60   60   61   63		
tt						

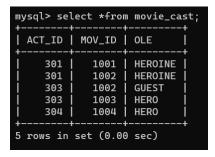
### Movie\_cast

mysql> CREATE TABLE MOVIE\_CAST (

- -> ACT\_ID INT (3),
- -> MOV\_ID INT (4),
- -> OLE VARCHAR (10),
- -> PRIMARY KEY (ACT\_ID, MOV\_ID),
- -> FOREIGN KEY (ACT\_ID) REFERENCES ACTOR (ACT\_ID),
- -> FOREIGN KEY (MOV\_ID) REFERENCES MOVIES (MOV\_ID)

->)

Field	Туре	Null	Key	Default	Extra
	int int varchar(10)	NO	•		   



### Rating

mysql> CREATE TABLE RATING (

- -> MOV\_ID INT (4),
- -> REV\_STARS VARCHAR (25),
- -> PRIMARY KEY (MOV\_ID),
- -> FOREIGN KEY (MOV\_ID) REFERENCES MOVIES (MOV\_ID)
- ->)

1. List the titles of all movies directed by 'Hitchcock'.

```
mysql> select mov_title from movies where dir_id = (select dir_id from director where dir_name = 'hitchcock');
+-----+
| mov_title |
+-----+
| AKASH |
+-----+
1 row in set (0.00 sec)
```

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

3. List all actors who acted in a movie before 2000 and also after 2015 (use JOIN operation).

4. Find the title of movies and number of stars for each movie that has at least 4 rating and find the highest number of stars that movie received. Sort the result by movie title.

5. Update rating of all movies directed by 'Steven Spielberg' to 5.

3. Design ERD for the following schema and execute the following Queries on it

#### **Students**

mysql> CREATE TABLE students (

- -> stno INT PRIMARY KEY,
- -> name VARCHAR(50),
- -> addr VARCHAR(255),
- -> city VARCHAR(50),

- -> state VARCHAR(2),
- -> zip VARCHAR(10)
- ->);

mysql> desc students;						
Field	Type	Null	Key	Default	Extra	
stno   name   addr   city   state   zip	int varchar(50) varchar(255) varchar(50) varchar(2) varchar(10)	NO YES YES YES YES YES	PRI	NULL NULL NULL NULL NULL		
t+ 6 rows in set (0.01 sec)						

## **Instructors**

mysql> CREATE TABLE INSTRUCTORS (

- -> empno INT PRIMARY KEY,
- -> name VARCHAR(50),
- -> ranks VARCHAR(20),
- -> roomno VARCHAR(10),
- -> telno VARCHAR(15)
- ->);

mysql> desc instructors;					
Field	Type	Null	Key	Default	Extra
name ranks roomno	int   varchar(50)   varchar(20)   varchar(10)   varchar(15)	YES	PRI	NULL NULL NULL NULL NULL	
5 rows in set (0.00 sec)					

# Courses

mysql> CREATE TABLE COURSES (

- -> cno INT PRIMARY KEY,
- -> cname VARCHAR(50),
- -> cr INT,
- -> cap INT
- ->)

Field	esc courses;  Type	+   Null	   Key	 Default	   Extra
cname cr	int varchar(50) int int	NO YES YES	PRI	NULL NULL NULL NULL	

## **Grades**

mysql> CREATE TABLE GRADES (

- -> stno INT,
- -> empno INT,
- -> cno INT,
- -> sem VARCHAR(10),
- -> year INT,
- -> grade INT,
- -> FOREIGN KEY (stno) REFERENCES students(stno),
- -> FOREIGN KEY (empno) REFERENCES INSTRUCTORS(empno),
- -> FOREIGN KEY (cno) REFERENCES COURSES(cno)
- ->);

mysql> desc grades;						
Field	Туре	Null	Key	Default	Extra	
sem   year	int int int varchar(10) int	NO YES YES YES YES YES YES		NULL NULL NULL NULL NULL		
++++++++ 6 rows in set (0.00 sec)						

# Advising

mysql> CREATE TABLE ADVISING (

- -> stno INT,
- -> empno INT,
- -> PRIMARY KEY (stno, empno),
- -> FOREIGN KEY (stno) REFERENCES students(stno),
- -> FOREIGN KEY (empno) REFERENCES INSTRUCTORS(empno)
- ->);

1. Find the names of students who took some four-credit course.

```
mysql> select students.stno,students.name,courses.cname from students inner join grades on students.stno = grades.stno i
nner join courses on grades.cno = courses.cno where courses.cr = 4;
              Edwards P. David |
mixon leatha |
                                               introduction to computing
  2661
                                               introduction to computing
                                              introduction to computing introduction to computing introduction to computing introduction to computing introduction to computing
  3566
5544
              pierce richard
              rawlings jerry |
Edwards P. David |
prior lorraine |
  1011
4022
  1011
2661
              Edwards P. David |
mixon leatha
                                               computer programming
                                              computer programming
   3566
              pierce richard
lewis jerry
prior lorraine
                                              computer programming
  5571
                                            | computer programming
| computer programming
  4022
11 rows in set (0.00 sec)
```

2. Find the names of students who took cs210 and cs310.

3. Find the names of students who took a course with an instructor who is also their advisor.

```
mysql> select students.stno,students.name,grades.empno,advising.empno from students inner join grades on students.stno =
grades.stno inner join advising on grades.stno = advising.stno where grades.empno = advising.empno;
   stno
              name
                                                              empno
                                                                  19
19
19
               Edwards P. David
Edwards P. David
                                                     19
                                                     19
19
   1011
               grogan a. mary
prior lorraine
lewis jerry
   2415
                                                   234
                                                                 234
234
   4022
   5571
5 rows in set (0.00 sec)
```

4. Find the names of all students whose advisor is not a full professor.

5. Find course numbers for courses that enroll exactly two students.

```
mysql> select courses.cno from courses inner join grades on courses.cno = grades.cno group by grades.cno having count(grades.stno)=2;
+----+
| cno |
+----+
| 410 |
| 310 |
+-----+
2 rows in set (0.00 sec)
```

6. Find the names of all students for whom no other student lives in the same city.

7. Find course numbers of courses taken by students who live in Boston and which are taught by an associate professor.

8. Find the telephone numbers of instructors who teach a course taken by any student who lives in Boston.

9. Find names of students who took every course taken by Will Samuel.

10. Find the names of the instructors who taught only one course during the spring semester of 2001.

11. Find the names of students who took only one course.

12. Find the names of instructors who teach no course.