

NAME: Shivam Sharma

SEC:2A

ROLL:56

ENROLL:12019009001262

Subject: PCC-CS 493 - Database Management Systems Lab

Date: 30/04/2021

1. Table/s creation; Value Insertion; Write the query; Show the Result; Draw

ER diagram of your database. (You may add more entities/relationship sets if required)

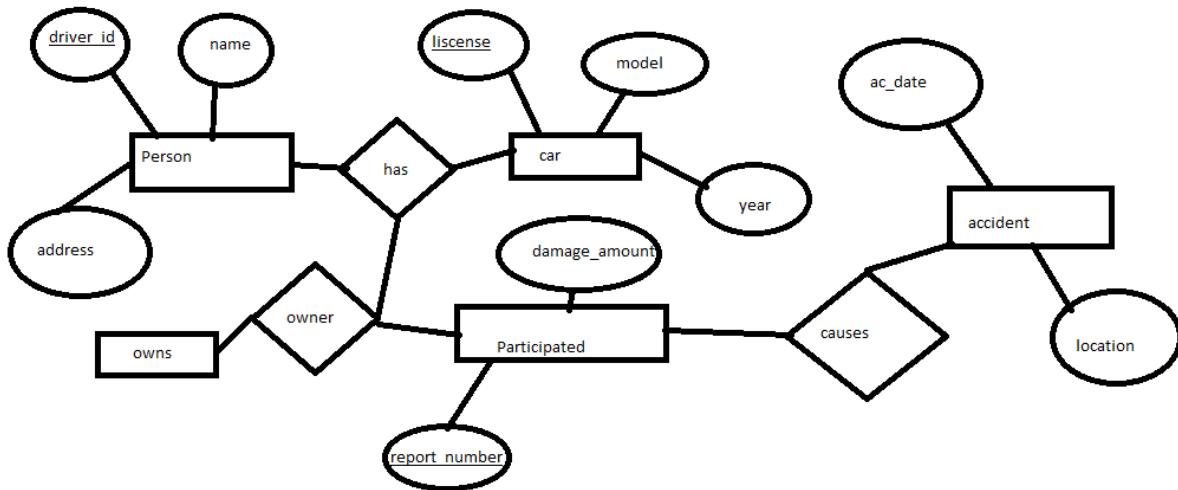


fig: ER diagram of Insurance database.

INSURANCE DATABASE:

Person (driver_id, name, address)

Car (license, model, year)

Accident (report_number, acdate, location)

Owns (driver_id, license)

Participated (driver_id, license, report_number, damage_amount)

Insert 5 records for each of above. (Read the queries carefully before insertion).

CREATE TABLE Person (

driver_id INTEGER NOT NULL PRIMARY KEY,
address VARCHAR2(20)
);

CREATE TABLE Car (

license NUMBER(10) NOT NULL PRIMARY KEY,
model INTEGER,
year DATE
);

CREATE TABLE Accident (

report_number NUMBER(6) PRIMARY KEY,
acdate DATE,
location VARCHAR2(20)
);

CREATE TABLE Owns (

driver_id INTEGER,
license NUMBER(10),
CONSTRAINT fk_owns_driver_id FOREIGN KEY (driver_id)
REFERENCES Person(driver_id),
CONSTRAINT fk_owns_license FOREIGN KEY (license)
REFERENCES Car(license)
);

insert into person values('D001','BABAI','TATANAGAR'),
('D002','KRIPTON','BIRSHANAGAR'),
('D003','SNEHA','KOLKATA'),

```
('D004','RAMAN','MIDNIPUR'),  
('D005','SHYAM','BARAKPOR');
```

	driver_id	name	address
▶	D001	BABAI	TATANAGAR
	D002	KRIPTON	BIRSHANAGAR
	D003	SNEHA	KOLKATA
	D004	RAMAN	MIDNIPUR
	D005	SHYAM	BARAKPOR
*	NULL	NULL	NULL

```
insert into car values('AABB2000','MARUTI 800',1998),  
('AACC3001','VAGANAR',2000),  
('AABC3200','SUV',1993),  
('CCDD2200','SADAN',1997),  
('AABC3000','SKODA',1995);
```

	license	model	c_year
▶	AABB2000	MARUTI 800	1998
	AABC3000	SKODA	1995
	AABC3200	SUV	1993
	AACC3001	VAGANAR	2000
	CCDD2200	SADAN	1997
*	NULL	NULL	NULL

```
insert into accident values('AR2197','1989-03-10','JAMSHEDPUR'),  
('AR2182','1990-12-23','KOLKATA'),  
('AR2134','1999-10-21','BIRSANAGAR'),  
('AR2153','1989-03-09','BARAKPOR'),  
('AR2123','1987-04-12','MADNIPOR');
```

Result Grid | Filter Rows:

	report_number	acdate	location
▶	AR2123	1987-04-12	MADNIPOR
	AR2134	1999-10-21	BIRSANAGAR
	AR2153	1989-03-09	BARAKPOR
	AR2182	1990-12-23	KOLKATA
	AR2197	1989-03-10	JAMSHPUR
*	NULL	NULL	NULL

```
insert into Owns values('D001','AABB2000'),
('D002','AABC3000'),
('D003','AABC3200'),
('D004','CCDD2200'),
('D005','AACC3001');
select * FROM Owns;
```

Result Grid | Filter

	driver_id	license
▶	D001	AABB2000
	D002	AABC3000
	D003	AABC3200
	D004	CCDD2200
	D005	AACC3001

```
insert into Participated values('D001','AABB2000','AR2197',2000),
('D002','AABC3000','AR2123',3000),
('D003','AABC3200','AR2182',2023),
('D004','CCDD2200','AR2153',1000),
('D005','AACC3001','AR2134',1200);
select * FROM Participated;
```

Result Grid | Filter Rows: Export:

	driver_id	license	report_number	damage_amount
▶	D001	AABB2000	AR2197	2000
	D002	AABC3000	AR2123	3000
	D003	AABC3200	AR2182	2023
	D004	CCDD2200	AR2153	1000
	D005	AACC3001	AR2134	1200

Construct the following SQL queries for this database.

a. Find the names of people who owned cars that were involved in accidents in 10-

03-1989.

```
select c.name from participated a join accident b on  
a.report_number=b.report_number and b.acdate='1989-03-10'  
join person c on a.driver_id=c.driver_id;
```

name
BABAI

1c)

```
update participated set damage_amount=3000 where  
report_number="AR2197" and license= "AABB2000";
```

driver_id	license	report_number	damage_amount
D001	AABB2000	AR2197	3000

1d)

```
SELECT * FROM exam.accident a join person b where a.location=b.address
```

report_number	acdate	location	driver_id	name	address
AR2182	1990-12-23	KOLKATA	D003	SNEHA	KOLKATA
AR2153	1989-03-09	BARAKPOR	D005	SHYAM	BARAKPOR

1e)

```
select datediff(sysdate(),acdate) from accident
```

datediff(sysdate(),acdate)
12437
7862
11740
11086
11739

SQL> create table author1(author_id int,name varchar(10),city varchar(10),country varchar(10),primary key(author_id));

Results Explain Describe Saved SQL History

Object Type TABLE Object AUTHOR1

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
AUTHOR1	AUTHOR1_ID	Number	-	-	0	1	-	-	-
	AUTHOR1_NAME	Varchar2	20	-	-	-	✓	-	-
	AUTHOR1_CITY	Varchar2	20	-	-	-	✓	-	-
	AUTHOR1_COUNTRY	Varchar2	20	-	-	-	✓	-	-

1 - 4

Language: en Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

SQL> create table publisher(publisher_id int,name varchar(10),city varchar(10),country varchar(10),primary key(publisher_id));

Results Explain Describe Saved SQL History

Object Type TABLE Object PUBLISHER1

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PUBLISHER1	PUBLISHER1_ID	Number	-	-	0	1	-	-	-
	PUBLISHER1_NAME	Varchar2	20	-	-	-	✓	-	-
	PUBLISHER1_CITY	Varchar2	20	-	-	-	✓	-	-
	PUBLISHER1_COUNTRY	Varchar2	20	-	-	-	✓	-	-

1 - 4

Language: en Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

16:25 23-04-2021

SQL> create table category(category_id int,description varchar(10),primary key(category_id));

Results Explain Describe Saved SQL History

Object Type TABLE Object CATEGORY1

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CATEGORY1	CATEGORY_ID	Number	-	-	0	1	-	-	-
	DESCRIPTION	Varchar2	30	-	-	-	✓	-	-

1 - 2

SQL> create table catalog1(book_id int,title varchar(10),author_id int,publisher_id int,category_id int,year int,price int,primary key(book_id),foreign key(author_id) references author(author_id),foreign key(publisher_id) references publisher(publisher_id),foreign key(category_id) references category(category_id));

Results Explain Describe Saved SQL History

Object Type TABLE Object CATALOGUE1

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CATALOGUE1	BOOK_ID	Number	-	-	0	1	-	-	-
	BOOK_TITLE	Varchar2	30	-	-	-	✓	-	-
	AUTHOR1_ID	Number	-	-	0	-	✓	-	-
	PUBLISHER1_ID	Number	-	-	0	-	✓	-	-
	CATEGORY_ID	Number	-	-	0	-	✓	-	-
	YEAR	Number	-	-	0	-	✓	-	-
	PRICE	Number	-	-	0	-	✓	-	-

1 - 7

SQL> create table order_details(order_no int,book_id int,quantity int,primary key(order_no,book_id),foreign key(book_id) references catalog(book_id));

Object Type TABLE Object ORDERDETAILS1										
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment	
ORDERDETAILS1	ORDER_ID	Number	-	-	0	1	-	-	-	
	BOOK_ID	Number	-	-	0	-	✓	-	-	
	QUANTITY	Number	-	-	0	-	✓	-	-	
1 - 3										

Language: en Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

ii) Enter at least five tuples for each relation.

SQL> INSERT INTO author
 (author1_id,author1_name,author1_city,author1_country) VALUES
 (1001,'JK Rowling','London','England'),
 (1002,'Chetan Bhagat','Mumbai','India'),
 (1003,'John McCarthy','Chicago','USA'),
 (1004,'Dan Brown','California','USA') ;

Results Explain Describe Saved SQL History			
AUTHOR1_ID	AUTHOR1_NAME	AUTHOR1_CITY	AUTHOR1_COUNTRY
1001	JK Rowling	London	England
1002	Chetan Bhagat	Mumbai	India
1003	John McCarthy	Chicago	USA
1004	Dan Brown	California	USA
1005	Mark Manson	California	USA

5 rows returned in 0.02 seconds [CSV Export](#) Language: en Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

SQL> INSERT INTO publisher1
 (publisher1_id,publisher1_name,publisher1_city,publisher1_country) VALUES
 (2001,'Bloomsbury','London','England'),
 (2002,'Scholastic','Washington','USA'),
 (2003,'Pearson','London','England'),
 (2004,'Rupa','Delhi','India') ;

Results Explain Describe Saved SQL History			
PUBLISHER1_ID	PUBLISHER1_NAME	PUBLISHER1_CITY	PUBLISHER1_COUNTRY
2001	Bloomsbury	London	England
2002	Scholastic	Washington	USA
2003	Pearson	London	England
2004	Rupa	Delhi	India
2005	Man	California	USA

5 rows returned in 0.00 seconds [CSV Export](#) Language: en Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

INSERT INTO category1 (category_id,description) VALUES
 (3001,'Fiction'),
 (3002,'Non-Fiction'),
 (3003,'thriller'),
 (3004,'action'),
 (3005,'fiction') ;

Results Explain Describe Saved SQL History

CATEGORY_ID	DESCRIPTION
3001	Fiction
3002	Non-Fiction
3003	thriller
3004	action
3005	fiction

5 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

```
SQL> INSERT INTO catalogue1 VALUES
(book_id,book_title,author1_id,publisher1_id,category_id,year,price)
(4001,'HP and Goblet Of Fire',1001,2001,3001,2002,600),
(4002,'HP and Order Of Phoenix',1001,2002,3001,2005,650),
(4003,'Two States',1002,2004,3001,2009,65),
(4004,'3 Mistakes of my life',1002,2004,3001,2007,55),
(4005,'Da Vinci Code',1004,2003,3001,2004,450),
(4006,'Angels and Demons',1004,2003,3001,2003,350),
(4007,'Artificial Intelligence',1003,2002,3002,1970,500) ;
```

Results Explain Describe Saved SQL History

BOOK_ID	BOOK_TITLE	AUTHOR1_ID	PUBLISHER1_ID	CATEGORY_ID	YEAR	PRICE
4001	HP and Goblet Of Fire	1001	2001	3001	2002	600
4002	HP and Order Of Phoenix	1001	2002	3001	2005	650
4003	Two States	1002	2004	3001	2009	65
4004	3 Mistakes of my life	1002	2004	3001	2007	55
4005	Da Vinci Code	1004	2003	3001	2004	450
4007	Artificial Intelligence	1003	2002	3002	1970	500

6 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

```
INSERT INTO orderdetails1 (order_id,book_id,quantity) VALUES
(5001,4001,5),
(5002,4002,7),
(5003,4003,15),
(5004,4004,11),
(5005,4005,9),
(5006,4006,8),
(5007,4007,2),
(5008,4004,3) ;
```

Results Explain Describe Saved SQL History

ORDER_ID	BOOK_ID	QUANTITY
5001	4001	5
5002	4002	7
5003	4003	15
5004	4004	11
5005	4005	9

5 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

iii) Give the details of the authors who have 2 or more books in the catalog and the price of the books is greater than the average price of the books in the catalog and the year of publication is after 2000.

```
SQL> SELECT * FROM author1
```

```

WHERE author1_id IN
(SELECT author1_id FROM catalogue1 WHERE
year>2000 AND price>
(SELECT AVG(price) FROM catalogue1)
GROUP BY author1_id HAVING COUNT(*)>1);

```

iv) Find the author of the book which has maximum sales.

```

mysql> SELECT author1_name FROM author1 a,catalogue1 c WHERE
a.author1_id=c.author1_id AND book_id IN (SELECT book_id FROM orderdetails1
WHERE quantity= (SELECT MAX(quantity) FROM orderdetails1));

```

2. Flights (flno (String), from, to, distance, depart_time, arrive_time, prices)

Aircraft (aid, fname, cruising_range)

Certified (eid, aid)

Employees (eid, ename, salary)

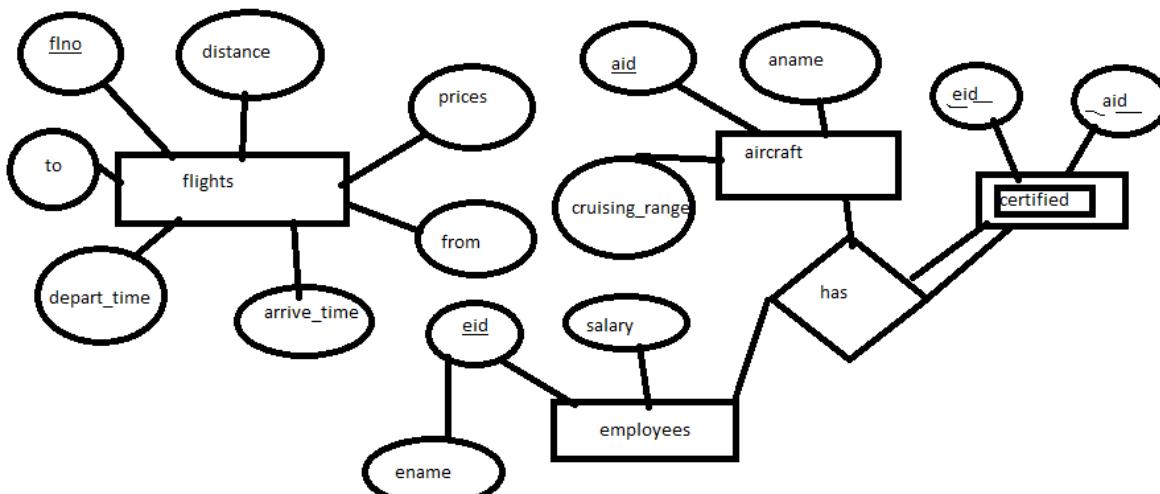


fig : ER diagram for flights and aircraft relationship table

```

CREATE TABLE flight(flno INT,frm VARCHAR(20),too
VARCHAR(20),distance INT,depart_time VARCHAR(20),arrive_time
VARCHAR(20),prices REAL,PRIMARY KEY (flno) );

```

```

CREATE TABLE aircraft(aid INT,fname VARCHAR(20),cruising_range
INT,PRIMARY KEY (aid) );

```

```

CREATE TABLE employees(eid INT,ename VARCHAR(20),salary
INT,PRIMARY KEY (eid) );

```

```
CREATE TABLE certified(eid INT,aid INT,PRIMARY KEY (eid,aid),FOREIGN KEY (eid) REFERENCES employees (eid),FOREIGN KEY (aid) REFERENCES aircraft (aid) );
```

```
INSERT INTO flight  
(flno,frm,too,distance,depart_time,arrive_time,prices) VALUES  
(1,'Bangalore','Mangalore',360,'10:45:00','12:00:00',10000);
```

```
INSERT INTO flight  
(flno,frm,too,distance,depart_time,arrive_time,prices) VALUES  
(2,'Bangalore','Delhi',5000,'12:15:00','04:30:00',25000);
```

```
INSERT INTO flight  
(flno,frm,too,distance,depart_time,arrive_time,prices) VALUES  
(3,'Bangalore','Mumbai',3500,'02:15:00','05:25:00',30000);
```

```
INSERT INTO flight  
(flno,frm,too,distance,depart_time,arrive_time,prices) VALUES  
(4,'Delhi','Mumbai',4500,'10:15:00','12:05:00',35000);
```

```
INSERT INTO flight  
(flno,frm,too,distance,depart_time,arrive_time,prices) VALUES  
(5,'Delhi','Frankfurt',18000,'07:15:00','05:30:00',90000);
```

```

SQL> desc flight;
Name                           Null?    Type
-----                         -----
FLNO                           NOT NULL NUMBER(38)
FRM                            VARCHAR2(20)
TOO                            VARCHAR2(20)
DISTANCE                        NUMBER(38)
DEPART_TIME                     VARCHAR2(20)
ARRIVE_TIME                     VARCHAR2(20)
PRICES                          FLOAT(63)

```

```
SQL> select * from flight
```

```
2
```

```
SQL> select * from flight;
```

FLNO	FRM	TOO	DISTANCE
DEPART_TIME		ARRIVE_TIME	PRICES
1	Bangalore	Mangalore	360
10:45:00		12:00:00	10000
2	Bangalore	Delhi	5000
12:15:00		04:30:00	25000
3	Bangalore	Mumbai	3500
02:15:00		05:25:00	30000

FLNO	FRM	TOO	DISTANCE
DEPART_TIME		ARRIVE_TIME	PRICES
4	Delhi	Mumbai	4500
10:15:00		12:05:00	35000
5	Delhi	Frankfurt	18000
07:15:00		05:30:00	90000

```

SQL> select * from aircraft;

      AID ANAME          CRUISING_RANGE
-----  -----
      1 jet                  10
      2 indigo                20
      3 goairways              30
      4 kingfisher              40
      5 kingfisher              50

```

```

SQL> select * from employees;

      EID ENAME          SALARY
-----  -----
      1 smith            20000
      2 jack             30000
      3 kingston         40000
      4 joe              50000
      5 julia            60000

```

```

SQL> select * from certified;

      EID      AID
-----  -----
      1          1
      2          2
      3          3
      4          4
      5          5

```

- 1. Find the minimum salary of that pilot among those pilots who are piloting same aircraft.**

```

mysql> SELECT eid, MIN(salary) AS salary
->   FROM employees
-> WHERE salary < (SELECT MAX(salary)
->                   FROM employees);
+-----+
| eid | salary |
+-----+
| 1  | 30000 |
+-----+
1 row in set (0.00 sec)

mysql>

```

- 2. Find the eids of employees who make the second highest salary.**

Select Max(Salary) as Salary from employees where Salary <(select MAX(Salary) from employees);

```
2
SQL> Select Max(Salary) as Salary from employees where Salary <(select MAX(Salary) from employees);

      SALARY
-----
      50000
```

3. Print the names of every non-pilot.

```
SQL> select ename from employees where eid not in(1,5);

ENAME
-----
jack
kingston
joe
```

4. Add a new field certification_date in the certified relation. How many days

are passed after certification?

```
SQL> alter table certified add certification_date date;

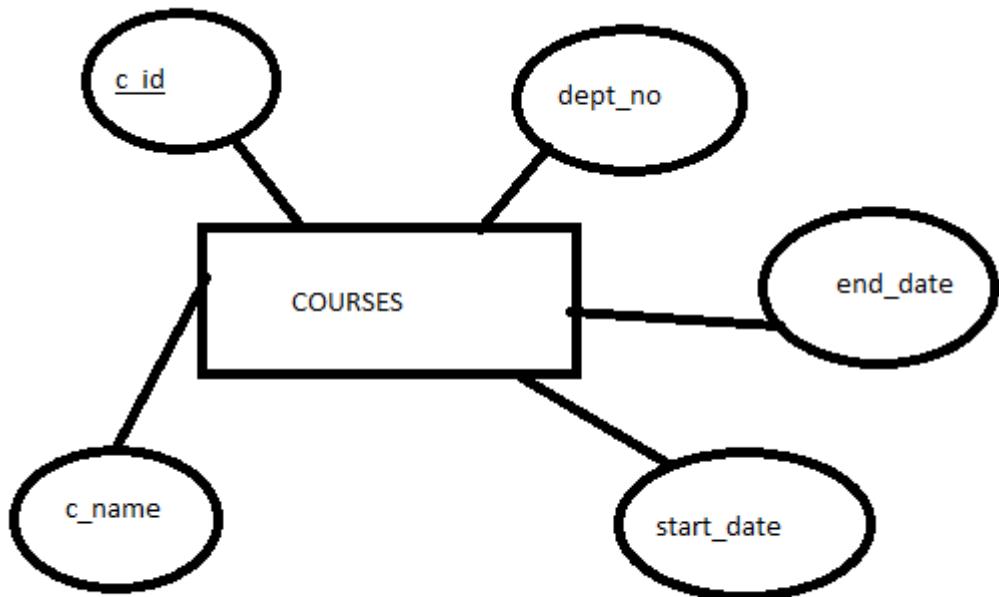
Table altered.
```

5. Add the character 'O' with the flightno whose price>1000;

```
mysql> select * from flight
-> ;
+-----+-----+-----+-----+-----+-----+-----+
| no | identification | frm | too | distance | departs | arrives | price |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | 0 | Bangalore | Mangalore | 360 | 10:45:00 | 12:00:00 | 10000 |
| 2 | 0 | Bangalore | Delhi | 5000 | 12:15:00 | 04:30:00 | 25000 |
| 3 | 0 | Bangalore | Mumbai | 3500 | 02:15:00 | 05:25:00 | 30000 |
| 4 | 0 | Delhi | Mumbai | 4500 | 10:15:00 | 12:05:00 | 35000 |
| 5 | 0 | Delhi | Frankfurt | 18000 | 07:15:00 | 05:30:00 | 90000 |
| 6 | 0 | Bangalore | Frankfurt | 19500 | 10:00:00 | 07:45:00 | 95000 |
| 7 | 0 | Bangalore | Frankfurt | 17000 | 12:00:00 | 06:30:00 | 99000 |
+-----+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

3. Create a table Courses (c_id, c_name, dept_no) to store records about 10

courses::



ER diagram of table "Courses" showing it's attribute and c_id as primary key.

```
create table courses(c_id number(4), c_name varchar2(20),dept_no number(5));
SQL> create table courses(c_id number(4), c_name varchar2(20),dept_no number(5));
Table created.
```

```
SQL> desc courses;
Name                           Null?    Type
-----                         -----
C_ID                          NUMBER(4)
C_NAME                         VARCHAR2(20)
DEPT_NO                        NUMBER(5)

SQL> select * from courses;
C_ID C_NAME                   DEPT_NO
---- -----
 1 btech                      1
 2 mtech                      2
 3 bca                        3
 4 mca                        4
 5 ba                          5
 6 ma                          6
 7 clerk                       7
 8 sales                       8
 9 primary_teacher              9
10 secondary_teacher            10

10 rows selected.
```

i) Add two columns start_date and end_date to the table.

Alter table courses add end_date date and start date date;

```

SQL> alter table courses add start_date date;
Table altered.

SQL> alter table courses add end_date date;
Table altered.

SQL> desc courses;
Name          Null?    Type
-----        -----
C_ID          NUMBER(4)
C_NAME        VARCHAR2(20)
DEPT_NO       NUMBER(5)
START_DATE    DATE
END_DATE      DATE

```

ii) Set c_id as the primary key for the table.

alter table courses add primary key(c_id);

```

SQL> alter table courses add primary key(c_id);
Table altered.

SQL> desc courses;
Name          Null?    Type
-----        -----
C_ID          NOT NULL NUMBER(4)
C_NAME        VARCHAR2(20)
DEPT_NO       NUMBER(5)
START_DATE    DATE
END_DATE      DATE

```

iii) Write a query to show the running courses within a given date range.

select c_name from courses where start_date between '1-jan-21' and '7-jan-21';

```

SQL> select c_name from courses where start_date between '1-jan-21' and '7-jan-21';
C_NAME
-----
btech
mtech
bca
mca

```

4. Solve the following queries

i. Find the ceiling and floor value of 14.887.

```
select ceil(14.887), floor(14.887) from dual;
```

```
SQL> select ceil(14.887), floor(14.887) from dual;  
CEIL(14.887) FLOOR(14.887)  
-----  
          15            14
```

ii) Find out the round-off 17.49989.

```
select round(17.49989) from dual;
```

```
SQL> select round(17.49989) from dual;  
ROUND(17.49989)  
-----  
          17
```
