1	Observe the following table and answer the parts(i) and(ii) accordingly **Table:Product**

Table:Product

Pno	Name	Qty	PurchaseDate
101	Pen	102	12-12-2011
102	Pencil	201	21-02-2013
103	Eraser	90	09-08-2010
109	Sharpener	90	31-08-2012
113	Clips	900	12-12-2011

Write the names of most appropriate columns, which can be considered as candidate keys.

What is the degree and cardinality of the above table?

Ans	Candidate Key: Pno, Name Degree:4 Cardinality:5

2	Write SQL queries for (i) to (iv) and find outputs for SQL queries (v) to (viii), which are based on the tables. **TRAINER**

TRAINER

TID	TNAME	CITY	HIREDATE	SALARY
101	SUNAINA	MUMBAI	1998-10-15	90000
102	ANAMIKA	DELHI	1994-12-24	80000
103	DEEPTI	CHANDIGARG	2001-12-21	82000
104	MEENAKSHI	DELHI	2002-12-25	78000
105	RICHA	MUMBAI	1996-01-12	95000
106	MANIPRABHA	CHENNAI	2001-12-12	69000

COURSE

CID	CNAME	FEES	STARTDATE	TID
C201	AGDCA	12000	2018-07-02	101
C202	ADCA	15000	2018-07-15	103
C203	DCA	10000	2018-10-01	102
C204	DDTP	9000	2018-09-15	104
C205	DHN	20000	2018-08-01	101
C206	O LEVEL	18000	2018-07-25	105

Display the Trainer Name, City & Salary in descending order of their Hiredate.

To display the TNAME and CITY of Trainer who joined the Institute in the month of December 2001.

To display TNAME, HIREDATE, CNAME, STARTDATE from tables TRAINER and COURSE of all those courses whose FEES is less than or equal to 10000.

To display number of Trainers from each city.

SELECT TID, TNAME, FROM TRAINER WHERE CITY NOT IN('DELHI', 'MUMBAI');

SELECT DISTINCT TID FROM COURSE;

SELECT TID, COUNT(*), MIN(FEES) FROM COURSE GROUP BY TID HAVING COUNT(*)>1;

SELECT COUNT(*), SUM(FEES) FROM COURSE WHERE STARTDATE< '2018-09-15';

Ans	SELECT TNAME, CITY, SALARY FROM TRAINER ORDER BY HIREDATE; SELECT TNAME, CITY FROM TRAINER WHERE HIREDATE BETWEEN '2001-12-01' AND '2001-12-31'; OR SELECT TNAME, CITY FROM TRAINER WHERE HIREDATE >= '2001-12-01' AND HIREDATE<='2001-12-31'; OR SELECT TNAME, CITY FROM TRAINER WHERE HIREDATE LIKE '2001-12%'; SELECT TNAME,HIREDATE,CNAME,STARTDATE FROM TRAINER, COURSE WHERE TRAINER.TID=COURSE.TID AND FEES<=10000; SELECT CITY, COUNT(*) FROM TRAINER GROUP BY CITY; TID TNAME 103 DEEPTI 106 MANIPRABHA DISTINCT TID 101 103 102

	104 105 TID COUNT(*) MIN(FEES) 101 2 12000 COUNT(*) SUM(FEES) 4 65000
3	Differentiate between DDL & DMLcommands. Identify DDL & DML commands from the following:- (UPDATE, SELECT, ALTER, DROP)
Ans	DDL stands for Data Definition language and comprises of commands which will change the structure of database object. DML stands for Data Manipulation Language and comprises of commands which are used to insert, edit, view & delete the data stored in a database object. DDL Commands: ALTER, DROP DML Commands: UPDATE, SELECT
4	Consider the following relations MobileMaster & MobileStock:-

MobileMaster

M_Id	M_Company	M_Name	M_Price	M_Mf_Date
MB001	Samsung	Galaxy	4500	2013-02-12
MB003	Nokia	N1100	2250	2011-04-15
MB004	Micromax	Unite3	4500	2016-10-17
MB005	Sony	XperiaM	7500	2017-11-20
MB006	Oppo	SelfieEx	8500	2010-08-21

MobileStock

S_Id	M_Id	M_Qty	M_Supplier
S001	MB004	450	New Vision
S002	MB003	250	Praveen Gallery
S003	MB001	300	Classic Mobile Store
S004	MB006	150	A-one Mobiles
S005	MB003	150	The Mobile
S006	MB006	50	Mobile Centre

Write the SQL query for questions from (i) to (iv) & write the output of SQL command for questions from (v) to (viii) given below:-

	(i) Display the Mobile company, Mobile name & price in descending order of their manufacturing date. (ii) List the details of mobile whose name starts with „S". (iii) Display the Mobile supplier & quantity of all mobiles except „MB003". (iv) To display the name of mobile company having price between 3000 & 5000. (v) SELECT M_Id, SUM(M_Qty) FROM MobileStock GROUP BY M_Id; (vi) SELECT MAX(M_Mf_Date), MIN(M_Mf_Date) FROM MobileMaster; (vii) SELECT M1.M_Id, M1.M_Name, M2.M_Qty, M2.M_Supplier FROM MobileMaster M1, MobileStock M2 WHERE M1.M_Id=M2.M_Id AND M2.M_Qty>=300; (viii) SELECT AVG(M_Price) FROM MobileMaster;
Ans	(i) SELECT M_Compnay, M_Name, M_Price FROM MobileMaster ORDER BY M_Mf_Date DESC; (½ mark for correct SELECT) (½ mark for correct ORDER BY) (ii) SELECT * FROM MobileMaster WHERE M_Name LIKE „S%"; (½ mark for correct SELECT) (½ mark for correct WHERE clause) (iii) SELECT M_Supplier, M_Qty FROM MobileStock WHERE M_Id <> „MB003"; (½ mark for correct SELECT) (½ mark for correct WHERE clause) (iv) SELECT M_Company FROM MobileMaster WHERE M_Price BETWEEN 3000 AND 5000; (½ mark for correct SELECT) (½ mark for correct BETWEEN clause) **(v)** <table><tr><td>M_Id</td><td>SUM(M_Qty)</td></tr><tr><td>MB004</td><td>450</td></tr><tr><td>MB003</td><td>400</td></tr><tr><td>MB001</td><td>300</td></tr><tr><td>MB006</td><td>200</td></tr></table> **(vi)** <table><tr><td>MAX(M_Mf_Date)</td><td>MIN(M_Mf_Date)</td></tr><tr><td>2017-11-20</td><td>2010-08-21</td></tr></table>

(vii)

M_Id	M_Name	M_Qty	M_Supplier
MB004	Unite3	450	New_Vision
MB001	Galaxy	300	Classic Mobile Store

5	Observe the following tables VIDEO and MEMBER carefully and write the name of the RDBMS operation out of (i) SELECTION (ii) PROJECTION (iii) UNION (iv) CARTESIAN PRODUCT, which has been used to produce the output as shown below, Also, find the Degree and Cardinality of the final result.

TABLE: VIDEO

VNO	VNAME	TYPE
F101	The Last Battle	Fiction
C101	Angels and Devils	Comedy
A102	Daredevils	Adventure

TABLE: MEMBER

MNO	MNAME
M101	Namish Gupta
M102	Sana Sheikh
M103	Lara James

FINAL RESULT

VNO	VNAME	TYPE	MNO	MNAME
F101	The Last Battle	Fiction	M101	Namish Gupta
F101	The Last Battle	Fiction	M102	Sana Sheikh
F101	The Last Battle	Fiction	M103	Lara James
C101	Angels and Devils	Comedy	M101	Namish Gupta
C101	Angels and Devils	Comedy	M102	Sana Sheikh
C101	Angels and Devils	Comedy	M103	Lara James
A102	Daredevils	Adventure	M101	Namish Gupta
A102	Daredevils	Adventure	M102	Sana Sheikh
A102	Daredevils	Adventure	M103	Lara James

Ans	CARTESIAN PRODUCT OR Option (iv) DEGREE = 5 CARDINALITY = 9

6	Write SQL queries for (i) to (iv) and find outputs for SQL queries (v) to (viii), which are based on the tables.

Table: ACCOUNT

ANO	ANAME	ADDRESS
101	Nirja Singh	Bangalore
102	Rohan Gupta	Chennai
103	Ali Reza	Hyderabad
104	Rishabh Jain	Chennai
105	Simran Kaur	Chandigarh

Table: TRANSACT

TRNO	ANO	AMOUNT	TYPE	DOT
T001	101	2500	Withdraw	2017-12-21
T002	103	3000	Deposit	2017-06-01
T003	102	2000	Withdraw	2017-05-12
T004	103	1000	Deposit	2017-10-22
T005	101	12000	Deposit	2017-11-06

Ans		
	(i)	To display details of all transactions of TYPE Deposit from Table TRANSACT.
	Ans	`SELECT * FROM TRANSACT WHERE TYPE = 'Deposit';`
	(ii)	To display the ANO and AMOUNT of all Deposits and Withdrawals done in the month of October 2017 from table TRANSACT.

	Ans	`SELECT ANO,AMOUNT FROM TRANSACT`
		`WHERE DOT >= '2017-10-01' AND DOT <= '2017-10-31';`
		OR
		`SELECT ANO,AMOUNT FROM TRANSACT`
		`WHERE DOT BETWEEN '2017-10-01' AND '2017-10-31';`

	(iii)	To display the last date of transaction (DOT) from the table TRANSACT for the Accounts having ANO as 103.
	Ans	`SELECT MAX(DOT) FROM TRANSACT WHERE ANO = 103;`

(iv)	To display all ANO, ANAME and DOT of those persons from tables ACCOUNT and TRANSACT who have done transactions less than or equal to 3000.

Ans	`SELECT ACCOUNT.ANO,ANAME,DOT FROM ACCOUNT,TRANSACT WHERE ACCOUNT.ANO=TRANSACT.ANO AND AMOUNT <=3000;`
	OR
	`SELECT A.ANO,ANAME,DOT FROM ACCOUNT A,TRANSACT T WHERE A.ANO=T.ANO AND AMOUNT <=3000;`

(v)	`SELECT ANO, ANAME FROM ACCOUNT` `WHERE ADDRESS NOT IN ('CHENNAI', 'BANGALORE');`

Ans	ANO	ANAME
	103	Ali Reza
	105	Simran Kaur

OR

ANO	ANAME
101	Nirja Singh
102	Rohan Gupta
103	Ali Reza
104	Rishabh Jain
105	Simran Kaur

(vi)	`SELECT DISTINCT ANO FROM TRANSACT;`

Ans	DISTINCT ANO
	101
	102
	103

(vii)	`SELECT ANO, COUNT(*), MIN(AMOUNT) FROM TRANSACT` `GROUP BY ANO HAVING COUNT(*)> 1;`

Ans	ANO	COUNT(*)	MIN(AMOUNT)
	101	2	2500
	103	2	1000

(viii)	`SELECT COUNT(*), SUM(AMOUNT) FROM TRANSACT` `WHERE DOT <= '2017-06-01';`

Ans	COUNT(*)	SUM(AMOUNT)
	2	5000

7	Observe the following tables BOOK and MEMBER carefully and write the name

of the RDBMS operation out of (i) SELECTION (ii) PROJECTION (iii) UNION (iv) CARTESIAN PRODUCT, which has been used to produce the output as shown in RESULT ? Also, find the Degree and Cardinality of the RESULT.

TABLE: BOOK

BNO	BNAME	TYPE
C101	Red Scorpion	Comic
F101	The Last Wish	Fiction
L102	French in a week	Literature

TABLE: MEMBER

MNO	MNAME
M101	Ritu Oberoi
M102	Fatima
M103	Jenny DeSouza

TABLE: RESULT

BNO	BNAME	TYPE	MNO	MNAME
C101	Red Scorpion	Comic	M101	Ritu Oberoi
C101	Red Scorpion	Comic	M102	Fatima
C101	Red Scorpion	Comic	M103	Jenny DeSouza
F101	The Last Wish	Fiction	M101	Ritu Oberoi
F101	The Last Wish	Fiction	M102	Fatima
F101	The Last Wish	Fiction	M103	Jenny DeSouza
L102	French in a week	Literature	M101	Ritu Oberoi
L102	French in a week	Literature	M102	Fatima
L102	French in a week	Literature	M103	Jenny DeSouza

| 8 | Write SQL queries for (i) to (iv) and find outputs for SQL queries (v) to (viii), which are based on the tables. |

TABLE:CUSTOMER

CNO	CNAME	ADDRESS
101	Richa Jain	Delhi
102	Surbhi Sinha	Chennai
103	Lisa Thomas	Bangalore
104	Imran Ali	Delhi
105	Roshan Singh	Chennai

TABLE : TRANSACTION

TRNO	CNO	AMOUNT	TYPE	DOT
T001	101	1500	Credit	2017-11-23
T002	103	2000	Debit	2017-05-12
T003	102	3000	Credit	2017-06-10
T004	103	12000	Credit	2017-09-12
T005	101	1000	Debit	2017-09-05

(i) To display details of all transactions of TYPE Credit from Table TRANSACTION.

(ii) To display the CNO and AMOUNT of all Transactions done in the month of September 2017 from table TRANSACTION.

(iii) To display the last date of transaction (DOT) from the table TRANSACTION for the customer having CNO as 103.

(iv) To display all CNO, CNAME and DOT (date of transaction) of those CUSTOMERS from tables CUSTOMER and TRANSACTION who have done transactions more than or equal to 2000.

(v) SELECT COUNT(*), AVG(AMOUNT) FROM TRANSACTION
WHERE DOT >= '2017-06-01';

(vi) SELECT CNO, COUNT(*), MAX (AMOUNT) FROM TRANSACTION
GROUP BY CNO HAVING COUNT(*)> 1;

(vii) SELECT CNO, CNAME FROM CUSTOMER
WHERE ADDRESS NOT IN ('DELHI', 'BANGALORE');

(viii) SELECT DISTINCT CNO FROM TRANSACTION;

9	Observe the following table MEMBER carefully and write the name of the RDBMS operation out of (i) SELECTION (ii) PROJECTION (iii) UNION (iv) CARTESIAN PRODUCT, which has been used to produce the output as shown in RESULT. Also, find the Degree and Cardinality of the RESULT :

MEMBER

NO	MNAME	STREAM
M001	JAYA	SCIENCE
M002	ADITYA	HUMANITIES
M003	HANSRAJ	SCIENCE
M004	SHIVAK	COMMERCE

RESULT

NO	MNAME	STREAM
M002	ADITYA	HUMANITIES

10	Write SQL queries for (i) to (iv) and find outputs for SQL queries (v) to (viii), which are based on the tables.

DVD

DCODE	DTITLE	DTYPE
F101	Henry Martin	Folk
C102	Dhrupad	Classical
C101	The Planets	Classical
F102	Universal Soldier	Folk
R102	A day in life	Rock

MEMBER

MID	NAME	DCODE	ISSUEDATE
101	AGAM SINGH	R102	2017-11-30
103	ARTH JOSEPH	F102	2016-12-13
102	NISHA HANS	C101	2017-07-24

(i) To display all details from the table MEMBER in descending order of ISSUEDATE.

(ii) To display the DCODE and DTITLE of all Folk Type DVDs from the table DVD.

(iii) To display the DTYPE and number of DVDs in each DTYPE from the table DVD.

(iv) To display all NAME and ISSUEDATE of those members from the table MEMBER who have DVDs issued (i.e., ISSUEDATE) in the year 2017.

(v) SELECT MIN(ISSUEDATE) FROM MEMBER;
(vi) SELECT DISTINCT DTYPE FROM DVD;
(vii) SELECT D.DCODE, NAME, DTITLE
FROM DVD D, MEMBER M WHERE D.DCODE=M.DCODE;
(viii) SELECT DTITLE FROM DVD
WHERE DTYPE NOT IN ("Folk", "Classical");

---

6 | Write SQL queries for (i) to (iv) and find outputs for SQL queries (v) to (viii), which are based on the tables.

Table: ACCOUNT

ANO	ANAME	ADDRESS
101	Nirja Singh	Bangalore
102	Rohan Gupta	Chennai
103	Ali Reza	Hyderabad
104	Rishabh Jain	Chennai
105	Simran Kaur	Chandigarh

Table: TRANSACT

TRNO	ANO	AMOUNT	TYPE	DOT
T001	101	2500	Withdraw	2017-12-21
T002	103	3000	Deposit	2017-06-01
T003	102	2000	Withdraw	2017-05-12
T004	103	1000	Deposit	2017-10-22
T005	101	12000	Deposit	2017-11-06

XII- Computer Science New (083)
Worksheet 3- SQL

Ans		
	(i)	To display details of all transactions of TYPE Deposit from Table TRANSACT.
	Ans	`SELECT * FROM TRANSACT WHERE TYPE = 'Deposit';`
	(ii)	To display the ANO and AMOUNT of all Deposits and Withdrawals done in the month of October 2017 from table TRANSACT.

	Ans	`SELECT ANO,AMOUNT FROM TRANSACT`
		`WHERE DOT >= '2017-10-01' AND DOT <= '2017-10-31';`
		OR
		`SELECT ANO,AMOUNT FROM TRANSACT`
		`WHERE DOT BETWEEN '2017-10-01' AND '2017-10-31';`

	(iii)	To display the last date of transaction (DOT) from the table TRANSACT for the Accounts having ANO as 103.
	Ans	`SELECT MAX(DOT) FROM TRANSACT WHERE ANO = 103;`
	(iv)	To display all ANO, ANAME and DOT of those persons from tables ACCOUNT and TRANSACT who have done transactions less than or equal to 3000.

	Ans	`SELECT ACCOUNT.ANO,ANAME,DOT FROM ACCOUNT,TRANSACT WHERE`
		`ACCOUNT.ANO=TRANSACT.ANO AND AMOUNT <=3000;`
		OR
		`SELECT A.ANO,ANAME,DOT FROM ACCOUNT A,TRANSACT T WHERE`
		`A.ANO=T.ANO AND AMOUNT <=3000;`

	(v)	`SELECT ANO, ANAME FROM ACCOUNT` `WHERE ADDRESS NOT IN ('CHENNAI', 'BANGALORE');`
	Ans	ANO ANAME 103 Ali Reza 105 Simran Kaur OR ANO ANAME 101 Nirja Singh 102 Rohan Gupta 103 Ali Reza 104 Rishabh Jain 105 Simran Kaur

(vi)	SELECT DISTINCT ANO FROM TRANSACT;
Ans	DISTINCT ANO 101 102 103
(vii)	SELECT ANO, COUNT(*), MIN(AMOUNT) FROM TRANSACT GROUP BY ANO HAVING COUNT(*)> 1;

Ans	ANO	COUNT(*)	MIN(AMOUNT)
	101	2	2500
	103	2	1000

(viii)	SELECT COUNT(*), SUM(AMOUNT) FROM TRANSACT WHERE DOT <= '2017-06-01';

Ans	COUNT(*)	SUM(AMOUNT)
	2	5000

---

7 Observe the following tables BOOK and MEMBER carefully and write the name of the RDBMS operation out of (i) SELECTION (ii) PROJECTION (iii) UNION (iv) CARTESIAN PRODUCT, which has been used to produce the output as shown in RESULT ? Also, find the Degree and Cardinality of the RESULT.

TABLE: BOOK

BNO	BNAME	TYPE
C101	Red Scorpion	Comic
F101	The Last Wish	Fiction
L102	French in a week	Literature

TABLE: MEMBER

MNO	MNAME
M101	Ritu Oberoi
M102	Fatima
M103	Jenny DeSouza

TABLE: RESULT

BNO	BNAME	TYPE	MNO	MNAME
C101	Red Scorpion	Comic	M101	Ritu Oberoi
C101	Red Scorpion	Comic	M102	Fatima
C101	Red Scorpion	Comic	M103	Jenny DeSouza
F101	The Last Wish	Fiction	M101	Ritu Oberoi
F101	The Last Wish	Fiction	M102	Fatima
F101	The Last Wish	Fiction	M103	Jenny DeSouza
L102	French in a week	Literature	M101	Ritu Oberoi
L102	French in a week	Literature	M102	Fatima
L102	French in a week	Literature	M103	Jenny DeSouza

8 Write SQL queries for (i) to (iv) and find outputs for SQL queries (v) to (viii), which are based on the tables.

TABLE: CUSTOMER

CNO	CNAME	ADDRESS
101	Richa Jain	Delhi
102	Surbhi Sinha	Chennai
103	Lisa Thomas	Bangalore
104	Imran Ali	Delhi
105	Roshan Singh	Chennai

TABLE: TRANSACTION

TRNO	CNO	AMOUNT	TYPE	DOT
T001	101	1500	Credit	2017-11-23
T002	103	2000	Debit	2017-05-12
T003	102	3000	Credit	2017-06-10
T004	103	12000	Credit	2017-09-12
T005	101	1000	Debit	2017-09-05

(i) To display details of all transactions of TYPE Credit from Table TRANSACTION.

(ii) To display the CNO and AMOUNT of all Transactions done in the month of September 2017 from table TRANSACTION.

(iii) To display the last date of transaction (DOT) from the table TRANSACTION for the customer having CNO as 103.

	(iv) To display all CNO, CNAME and DOT (date of transaction) of those CUSTOMERS from tables CUSTOMER and TRANSACTION who have done transactions more than or equal to 2000. (v) SELECT COUNT(*), AVG(AMOUNT) FROM TRANSACTION WHERE DOT >= '2017-06-01'; (vi) SELECT CNO, COUNT(*), MAX (AMOUNT) FROM TRANSACTION GROUP BY CNO HAVING COUNT(*)> 1; (vii) SELECT CNO, CNAME FROM CUSTOMER WHERE ADDRESS NOT IN ('DELHI', 'BANGALORE'); (viii) SELECT DISTINCT CNO FROM TRANSACTION;
9	Observe the following table MEMBER carefully and write the name of the RDBMS operation out of (i) SELECTION (ii) PROJECTION (iii) UNION (iv) CARTESIAN PRODUCT, which has been used to produce the output as shown in RESULT. Also, find the Degree and Cardinality of the RESULT : **MEMBER** <table><tr><td>NO</td><td>MNAME</td><td>STREAM</td></tr><tr><td>M001</td><td>JAYA</td><td>SCIENCE</td></tr><tr><td>M002</td><td>ADITYA</td><td>HUMANITIES</td></tr><tr><td>M003</td><td>HANSRAJ</td><td>SCIENCE</td></tr><tr><td>M004</td><td>SHIVAK</td><td>COMMERCE</td></tr></table> **RESULT** <table><tr><td>NO</td><td>MNAME</td><td>STREAM</td></tr><tr><td>M002</td><td>ADITYA</td><td>HUMANITIES</td></tr></table>
10	Write SQL queries for (i) to (iv) and find outputs for SQL queries (v) to (viii), which are based on the tables.

DVD

DCODE	DTITLE	DTYPE
F101	Henry Martin	Folk
C102	Dhrupad	Classical
C101	The Planets	Classical
F102	Universal Soldier	Folk
R102	A day in life	Rock

MEMBER

MID	NAME	DCODE	ISSUEDATE
101	AGAM SINGH	R102	2017-11-30
103	ARTH JOSEPH	F102	2016-12-13
102	NISHA HANS	C101	2017-07-24

(i) To display all details from the table MEMBER in descending order of ISSUEDATE.
(ii) To display the DCODE and DTITLE of all Folk Type DVDs from the table DVD.
(iii) To display the DTYPE and number of DVDs in each DTYPE from the table DVD.
(iv) To display all NAME and ISSUEDATE of those members from the table MEMBER who have DVDs issued (i.e., ISSUEDATE) in the year 2017.
(v) SELECT MIN(ISSUEDATE) FROM MEMBER;
(vi) SELECT DISTINCT DTYPE FROM DVD;
(vii) SELECT D.DCODE, NAME, DTITLE
FROM DVD D, MEMBER M WHERE D.DCODE=M.DCODE;
(viii) SELECT DTITLE FROM DVD
WHERE DTYPE NOT IN ("Folk", "Classical");

11	Observe the following table CANDIDATE carefully and write the name of the RDBMS operation out of (i) SELECTION (ii) PROJECTION (iii) UNION (iv) CARTESIAN PRODUCT, which has been used to produce the output as shown in RESULT. Also, find the Degree and Cardinality of the RESULT.

TABLE : CANDIDATE

NO	NAME	STREAM
C1	AJAY	LAW
C2	ADITI	MEDICAL
C3	ROHAN	EDUCATION
C4	RISHAV	ENGINEERING

RESULT

NO	NAME
C3	ROHAN

---

12 Write SQL queries for (i) to (iv) and find outputs for SQL queries (v) to (viii), which are based on the tables :

Code	BNAME	TYPE
F101	The priest	Fiction
L102	German easy	Literature
C101	Tarzan in the lost world	Comic
F102	Untold Story	Fiction
C102	War heroes	Comic

TABLE : MEMBER

MNO	MNAME	CODE	ISSUEDATE
M101	RAGHAV SINHA	L102	2016-10-13
M103	SARTHAK JOHN	F102	2017-02-23
M102	ANISHA KHAN	C101	2016-06-12

(i) To display all details from table MEMBER in descending order of ISSUEDATE.
(ii) To display the BNO and BNAME of all Fiction Type books from the table BOOK.
(iii) To display the TYPE and number of books in each TYPE from the table BOOK.
(iv) To display all MNAME and ISSUEDATE of those members from table MEMBER who have books issued (i.e. ISSUEDATE) in the year 2017.
(v) **SELECT MAX(ISSUEDATE) FROM MEMBER;**

	(vi) **SELECT DISTINCT TYPE FROM BOOK;** (vii) **SELECT A.CODE,BNAME,MNO,MNAME** **FROM BOOK A, MEMBER B WHERE A.CODE=B.CODE;** (viii) **SELECT BNAME FROM BOOK** **WHERE TYPE NOT IN ("FICTION","COMIC");**
13	Observe the table 'Club' given below: **Club** 	Member_id	Member_Name	Address	Age	Fee
---	---	---	---	---
M001	Sumit	New Delhi	20	2000
M002	Nisha	Gurgaon	19	3500
M003	Niharika	New Delhi	21	2100
M004	Sachin	Faridabad	18	3500	 i. What is the cardinality and degree of the above given table? ii. If a new column contact_no has been added and three more members have joined the club then how these changes will affect the degree and cardinality of the above given table.
Ans	i. Cadinality: 4 Degree: 5 ii. Cardinality: 7 Degree: 6
14	Write SQL commands for the queries (i) to (iv) and output for (v) to (viii) based on the tables 'Watches' and Sale given below. **Watches** 	Watchid	Watch_Name	Price	Type	Qty_Store
---	---	---	---	---
W001	HighTime	10000	Unisex	100
W002	LifeTime	15000	Ladies	150
W003	Wave	20000	Gents	200
W004	HighFashion	7000	Unisex	250
W005	GoldenTime	25000	Gents	100 | **Sale** | Watchid	Qty_Sold	Quarter
---	---	---
W001	10	1
W003	5	1
W002	20	2
W003	10	2
W001	15	3
W002	20	3
W005	10	3
W003	15	4	

	i. To display all the details of those watches whose name ends with 'Time' ii. To display watch's name and price of those watches which have price range in between 5000-15000. iii. To display total quantity in store of Unisex type watches. iv. To display watch name and their quantity sold in first quarter. v. select max(price), min(qty_store) from watches; vi. select quarter, sum(qty_sold) from sale group by quarter; vii. select watch_name,price,type from watches w, sale s where w.watchid!=s.watchid; viii. select watch_name, qty_store, sum(qty_sold), qty_storesum(qty_sold) "Stock" from watches w, sale s where w.watchid=s.watchid group by s.watchid;
Ans	**i. select * from watches where watch_name like '%Time'** **ii. select watch_name, price from watches where price between 5000 and 15000;** **iii. select sum(qty_store) from watches where type like 'Unisex';** **iv. select watch_name,qty_sold from watches w,sale s where w.watchid=s.watchid and quarter=1;** v. 	max(price)	min(qty_store)
25000	100	 vi. 	quarter	sum(qty_sold)
1	15
2	30
3	45
4	15	 vii. 	watch_name	price	type
---	---	---
HighFashion	7000	Unisex | Viii | watch_name	qty_store	qty_sold	Stock
---	---	---	---
HighTime	100	25	75
LifeTime	150	40	110
Wave	200	30	170
GoldenTime	100	10	90	
15	Differentiate between cardinality and degree of a table with the help of an example.
Ans	Cardinality is defined as the number of rows in a table.

	Degree is the number of columns in a table.
	Eg:Consider the following tables:
	Table : Account

Acno	Cname
Ac100	Sheela
Ac101	Darsh
Ac102	Kathy

Cardinality of Account table is : 3
Degree of Account table is :2

---

16 Consider the following tables FACULTY and COURSES. Write SQL commands for the statements (i) to (v) and give outputs for SQL queries (vi) to (vii)

FACULTY

F_ID	Fname	Lname	Hire_date	Salary
102	Amit	Mishra	12-10-1998	12000
103	Nitin	Vyas	24-12-1994	8000
104	Rakshit	Soni	18-5-2001	14000
105	Rashmi	Malhotra	11-9-2004	11000
106	Sulekha	Srivastava	5-6-2006	10000

COURSES

C_ID	F_ID	Cname	Fees
C21	102	Grid Computing	40000
C22	106	System Design	16000
C23	104	Computer Security	8000
C24	106	Human Biology	15000
C25	102	Computer Network	20000
C26	105	Visual Basic	6000

i) To display details of those Faculties whose salary is greater than 12000.
where salary > 12000
ii) To display the details of courses whose fees is in the range of 15000 to 50000 (both values included).
iii) To increase the fees of all courses by 500 of "System Design" Course.
iv) To display details of those courses which are taught by 'Sulekha' in descending order of
courses.
Select COUNT(DISTINCT F_ID) from COURSES;
v) Select COUNT(DISTINCT F_ID) from COURSES;

	vi) Select MIN(Salary) from FACULTY,COURSES where COURSES.F_ID = FACULTY.F_ID;
Ans	Ans: Select * from faculty Ans: Select * from Courses .where fees between 15000 and 50000 Ans: Update courses set fees = fees + 500 where Cname = "System Design" Ans: Select * from faculty fac,courses cour where fac.f_id = cour.f_id and fac.fname = 'Sulekha' order by cname desc Ans: 4 Ans: 6000