2. Create the following tables and perform the queries that follow.

College

Column Name	Data type	Size
cName	varchar2	10
state	varchar2	10
enrollment	int	

Student

Column Name	Data type	Size
sID	int	
sName	varchar2	10
GPA	number	2,1
sizeHS	int	
DoB	date	

Apply

Column Name	Data type	Size
sID	int	
cName	varchar2	10
major	varchar2	20
decision	char	1

College

CNAME	STATE	ENROLLMENT
Stanford	CA	15000
Berkeley	CA	36000
MIT	MA	10000
Cornell	NY	21000
Harvard	MA	50040
Stanford	CA	15000
Berkeley	CA	36000
MIT	MA	10000
Cornell	NY	21000
Harvard	MA	50040
BITS GOA	GA	10012

Student

SID	SNAME	GPA	SIZEHS	DOB
122	Amy	3.9	1000	26-JUN-96
234	Bob	3.6	1500	04-APR-95
345	Craig	3.5	500	04-FEB-95
456	Doris	3.9	1000	24-JUL-97
567	Edward	2.9	2000	21-DEC-96
678	Fay	3.8	200	27-AUG-96
789	Gray	3.4	800	08-OCT-96
987	Helen	3.7	800	27-MAR-97
876	Irene	3.9	400	07-MAR-96
765	Jay	2.9	1500	08-AUG-98
654	Amy	3.9	1000	26-MAY-96
543	Craig	3.4	2000	27-MAY-96

Apply

SID	CNAME	MAJOR	DECISION
123	Stanford	CS	Y
123	Stanford	EE	N
123	Berkeley	CS	Y
123	Cornell	EE	Y
234	Berkeley	biology	N
345	MIT	bioengineering	Y
345	Cornell	bioengineering	N
345	Cornell	CS	Y
345	Cornell	EE	N
678	Stanford	hitory	Y
987	Stanford	CS	Y
987	Berkeley	CS	Y
876	Stanford	CS	N
765	Stanford	history	Y
765	Cornell	history	N
765	Cornell	psychology	Y
543	MIT	CS	N

a) List the student name, dob from student table.

```
1  SELECT SNAME, DOB FROM STUDENT;
```

SNAME	DOB
Amy	26-JUN-96
Bob	04-APR-95
Craig	04-FEB-95
Doris	24-JUL-97
Edward	21-DEC-96
Fay	27-AUG-96
Gray	08-OCT-96
Helen	27-MAR-97
Irene	07-MAR-96
Jay	08-AUG-98
Amy	26-MAY-96
Craig	27-MAY-96

Download CSV

b) List the name of students scoring more than 3.7 in GPA.

```
1  SELECT SNAME FROM STUDENT WHERE GPA > 3.7;
```

SNAME
Amy
Doris
Fay
Irene
Amy

Download CSV
5 rows selected.

c) List the name of students whose High School size is at least 1000 and born after 1996. [Hint: check DoB greater than 31st December, 1996]

```
1  SELECT SNAME FROM STUDENT WHERE SIZEHS >=1000 AND DOB >'31-DEC-1996';
```

SNAME
Doris
Jay

Download CSV
2 rows selected.

d) **List the name of students who are scoring GPA in between 2.9 and 3.9.**

SQL Worksheet   Clear   Find   Actions ✓   Save   Run ▶

```
1   SELECT SNAME FROM STUDENT WHERE GPA > 2.9 AND GPA < 3.9;
```

SNAME
Bob
Craig
Fay
Gray
Helen
Craig

Download CSV
6 rows selected.

e) **List all the details of colleges who situated in Goa.**

SQL Worksheet   Clear   Find   Actions ✓   Save   Run ▶

```
1   SELECT * FROM COLLEGE WHERE STATE = 'GA';
```

CNAME	STATE	ENROLLMENT
BITS GOA	GA	10012

Download CSV

f) **List the students who have scored more than 2.0 but less than 3.5.**

SQL Worksheet   Clear   Find   Actions ✓   Save   Run ▶

```
1   SELECT * FROM STUDENT WHERE GPA > 2.0 AND GPA < 3.5
```

SID	SNAME	GPA	SIZEHS	DOB
567	Edward	2.9	2000	21-DEC-96
789	Gray	3.4	800	08-OCT-96
765	Jay	2.9	1500	08-AUG-98
543	Craig	3.4	2000	27-MAY-96

Download CSV
4 rows selected.

g) List the students who were born after 1st Jul 96 in the order of the Date of Birth.

SQL Worksheet　　　　　　　　　　　　　　Clear　　Find　　Actions ∨　　Save　　**Run ▶**

```
1  SELECT * FROM STUDENT WHERE DOB > '1-JUL-1996' ORDER BY DOB;
```

SID	SNAME	GPA	SIZEHS	DOB
678	Fay	3.8	200	27-AUG-96
789	Gray	3.4	800	08-OCT-96
567	Edward	2.9	2000	21-DEC-96
987	Helen	3.7	800	27-MAR-97
456	Doris	3.9	1000	24-JUL-97
765	Jay	2.9	1500	08-AUG-98

Download CSV
6 rows selected.

h) List the sID, cName, decision of applications that are accepted.

SQL Worksheet　　　　　　　　　　　　　　Clear　　Find　　Actions ∨　　Save　　**Run ▶**

```
1  SELECT SID, CNAME, DECISION FROM APPLY WHERE DECISION = 'Y'
```

SID	CNAME	DECISION
123	Stanford	Y
123	Berkeley	Y
123	Cornell	Y
345	MIT	Y
345	Cornell	Y
678	Stanford	Y
987	Stanford	Y
987	Berkeley	Y
765	Stanford	Y
765	Cornell	Y

Download CSV
10 rows selected.

i) List the colleges that has enrolment greater than 10001.

SQL Worksheet　　　　　　　　　　　　　　Clear　　Find　　Actions ∨　　Save　　**Run ▶**

```
1  SELECT CNAME FROM COLLEGE WHERE ENROLLMENT > 10001
```

CNAME
Stanford
Berkeley
Cornell
Harvard
Stanford
Berkeley
Cornell
Harvard
BITS GOA

Download CSV
9 rows selected.

j) **List the colleges not in California.**

```
1   SELECT CNAME FROM COLLEGE WHERE STATE != 'CA'
```

CNAME
MIT
Cornell
Harvard
MIT
Cornell
Harvard
BITS GOA

Download CSV
7 rows selected.

k) **List names of all student who came from high school having size greater than 1700 and scored GPA less than 3.8.**

```
1   SELECT SNAME FROM STUDENT WHERE SIZEHS > 1700 AND GPA < 3.8
```

SNAME
Edward
Craig

Download CSV
2 rows selected.

l) **Display the description of the Student table.**

```
1   DESC STUDENT
```

TABLE STUDENT

Column	Null?	Type
SID	-	NUMBER
SNAME	-	VARCHAR2(20)
GPA	-	NUMBER(3,1)
SIZEHS	-	NUMBER
DOB	-	DATE

Download CSV
5 rows selected.

m) Display the details of all students.

```
1  SELECT* FROM STUDENT
2
```

SID	SNAME	GPA	SIZEHS	DOB
122	Amy	3.9	1000	26-JUN-96
234	Bob	3.6	1500	04-APR-95
345	Craig	3.5	500	04-FEB-95
456	Doris	3.9	1000	24-JUL-97
567	Edward	2.9	2000	21-DEC-96
678	Fay	3.8	200	27-AUG-96
789	Gray	3.4	800	08-OCT-96
987	Helen	3.7	800	27-MAR-97
876	Irene	3.9	400	07-MAR-96
765	Jay	2.9	1500	08-AUG-98
654	Amy	3.9	1000	26-MAY-96
543	Craig	3.4	2000	27-MAY-96

Download CSV
12 rows selected.

n) Display unique majors.

```
1  SELECT DISTINCT MAJOR FROM APPLY
2
```

MAJOR
psychology
EE
biology
hitory
history
bioengineering
CS

Download CSV
7 rows selected.

o) List the student names those are having five characters in their Names.

```
1  SELECT SNAME FROM STUDENT WHERE SNAME LIKE '_____'
2
```

SNAME
Craig
Doris
Helen
Irene
Craig

Download CSV
5 rows selected.

p) List the student names those are starting with 'H' and with five characters.

```
1  SELECT SNAME FROM STUDENT WHERE SNAME LIKE 'H____'
2
```

SNAME
Helen

Download CSV

q) List the student names those are having third character and fifth char as 'e'.

SQL Worksheet Clear Find Actions ⌄ Save Run ▶

```
1  SELECT SNAME FROM STUDENT WHERE SNAME LIKE '__e_e%'
2
```

SNAME
Irene

Download CSV

r) List the student names ending with 'y'

SQL Worksheet Clear Find Actions ⌄ Save Run ▶

```
1  SELECT SNAME FROM STUDENT WHERE SNAME LIKE '%y'
2
```

SNAME
Amy
Fay
Gray
Jay
Amy

Download CSV
5 rows selected.

s) List the Students in the order of their GPA.

SQL Worksheet Clear Find Actions ⌄ Save Run ▶

```
1  SELECT SNAME FROM STUDENT ORDER BY GPA DESC;
2
```

SNAME
Amy
Amy
Irene
Doris
Fay
Helen
Bob
Craig
Craig
Gray
Jay
Edward

Download CSV
12 rows selected.

t) List the details of the students in order of the ascending of GPA and descending of DoB.

SQL Worksheet   Clear   Find   Actions ▾   Save   Run ▶

```
1   SELECT * FROM STUDENT ORDER BY GPA ASC, DOB DESC
```

SID	SNAME	GPA	SIZEHS	DOB
765	Jay	2.9	1500	08-AUG-98
567	Edward	2.9	2000	21-DEC-96
789	Gray	3.4	800	08-OCT-96
543	Craig	3.4	2000	27-MAY-96
345	Craig	3.5	500	04-FEB-95
234	Bob	3.6	1500	04-APR-95
987	Helen	3.7	800	27-MAR-97
678	Fay	3.8	200	27-AUG-96
456	Doris	3.9	1000	24-JUL-97
122	Amy	3.9	1000	26-JUN-96
654	Amy	3.9	1000	26-MAY-96
876	Irene	3.9	400	07-MAR-96

Download CSV
12 rows selected.

u) List the sIDs of student who apply in either 'Harvard' and 'Berkeley' college.

SQL Worksheet   Clear   Find   Actions ▾   Save   Run ▶

```
1   SELECT SID FROM APPLY WHERE CNAME = 'Harvard' OR CNAME = 'Berkeley'
```

SID
123
234
987

Download CSV
3 rows selected.

v) Delete all applications filled at Stanford (Choose table wisely)

SQL Worksheet   Clear   Find   Actions ▾   Save   Run ▶

```
1   DELETE FROM APPLY WHERE CNAME = 'Stanford'
```

6 row(s) deleted.

w) Delete the college Stanford from college table.

SQL Worksheet   Clear   Find   Actions ▾   Save   Run ▶

```
1   DELETE FROM COLLEGE WHERE CNAME = 'Stanford'
```

2 row(s) deleted.

x) **Modify the GPA of all students by giving 10% raise in their GPA.**

SQL Worksheet   ◇ Clear   ✎ Find   Actions ∨   💾 Save   Run ▶

```
1   UPDATE STUDENT SET GPA=GPA+(GPA*0.10);
```

```
12 row(s) updated.
```

y) **Increment the GPA of the students by 1.5 whose GPA is less than 3.5 and belongs to High School having size greater than 1500.**

SQL Worksheet   ◇ Clear   ✎ Find   Actions ∨   💾 Save   Run ▶

```
1   UPDATE STUDENT SET GPA = GPA+1.5 WHERE GPA < 3.5 AND SIZEHS > 1500
```

```
1 row(s) updated.
```

z) **Delete the students who have scored less than 3.2 GPA.**

SQL Worksheet   ◇ Clear   ✎ Find   Actions ∨   💾 Save   Run ▶

```
1   DELETE FROM STUDENT WHERE GPA < 4.0
```

```
4 row(s) deleted.
```

3. Execute the following queries:
Consider the following schema:
candidate (candidate_id, candidate_name, class, city, fee, DoB)

- Create a table named candidate where candidate_id should be a primary key

SQL Worksheet   Clear   Find   Actions ⌄   Save   Run ▶

```
 1  create table candidate(
 2  candidate_id int PRIMARY KEY,
 3  candidate_name varchar2(20),
 4  class varchar2(20),
 5  city varchar2(20),
 6  fee int,
 7  dob date
 8  );
 9
10  describe candidate
```

TABLE CANDIDATE

Column	Null?	Type
CANDIDATE_ID	NOT NULL	NUMBER
CANDIDATE_NAME	-	VARCHAR2(20)
CLASS	-	VARCHAR2(20)
CITY	-	VARCHAR2(20)
FEE	-	NUMBER
DOB	-	DATE

Download CSV
6 rows selected.

- Insert 5 tuples into table.

SQL Worksheet   Clear   Find   Actions ⌄   Save   Run ▶

```
 1  select * from candidate;
```

CANDIDATE_ID	CANDIDATE_NAME	CLASS	CITY	FEE	DOB
1	Unzila	3rd year	Meerut	110188	01-JUN-02
2	Karan	2nd year	Pune	131100	06-FEB-02
3	Ananya	1st year	Noida	125000	21-JAN-04
4	Kamlesh	3rd year	Dehradun	120100	29-DEC-02
5	Samya	3rd year	Karnal	120100	04-AUG-02
6	Imad	2nd year	Meerut	125000	21-DEC-03
7	Anant	4th year	Banglore	145000	11-SEP-00
8	Tanishka	4th year	Rishikesh	150000	05-APR-01
9	Vivek	3rd year	Srinagar	135000	14-DEC-01
10	Naman	1st year	Delhi	130200	25-MAY-03

- Display the candidate names and class who either belong to Dehradun or Srinagar.

SQL Worksheet ⬧ Clear ✎ Find Actions ⌄ 💾 Save Run ▶

```
1  select candidate_name, class from candidate where city='Dehradun' or city='Srinagar';
```

CANDIDATE_NAME	CLASS
Kamlesh	3rd year
Vivek	3rd year

Download CSV
2 rows selected.

- Display the candidate_id whose fee lies between 50000 – 85000.

SQL Worksheet ⬧ Clear ✎ Find Actions ⌄ 💾 Save Run ▶

```
1   select candidate_id from candidate where fee between 100000 and 125000;
```

CANDIDATE_ID
1
3
4
5
6

Download CSV
5 rows selected.

- Find the candidate details who were born after 10-Jan-2000 in descending order.

SQL Worksheet ⬧ Clear ✎ Find Actions ⌄ 💾 Save Run ▶

```
1  select * from candidate where dob>'10-JAN-2001' order by dob desc;
```

CANDIDATE_ID	CANDIDATE_NAME	CLASS	CITY	FEE	DOB
3	Ananya	1st year	Noida	125000	21-JAN-04
6	Imad	2nd year	Meerut	125000	21-DEC-03
10	Naman	1st year	Delhi	130200	25-MAY-03
4	Kamlesh	3rd year	Dehradun	120100	29-DEC-02
5	Samya	3rd year	Karnal	120100	04-AUG-02
1	Unzila	3rd year	Meerut	110188	01-JUN-02
2	Karan	2nd year	Pune	131100	06-FEB-02
9	Vivek	3rd year	Srinagar	135000	14-DEC-01
8	Tanishka	4th year	Rishikesh	150000	05-APR-01

Download CSV
9 rows selected.

- Remove the details of the candidate whose name contains 'b' and ends with 'k'.

SQL Worksheet ⬧ Clear ✎ Find Actions ⌄ 💾 Save Run ▶

```
1  delete from candidate where candidate_name like '%a%n';
```

2 row(s) deleted.

- Increase and display the fee all candidates of class '3ʳᵈ year' by 10%.

SQL Worksheet   Clear   Find   Actions ∨   Save   Run ▶

```
1  select (fee+fee*0.1) as Updated_fee from candidate where class='3rd year';
```

UPDATED_FEE
121206.8
132110
132110
148500

Download CSV
4 rows selected.

- Add a new column 'GPA' to the existing table.

SQL Worksheet   Clear   Find   Actions ∨   Save   Run ▶

```
1  alter table candidate add GPA number(2,1);
2  select * from candidate;
```

CANDIDATE_ID	CANDIDATE_NAME	CLASS	CITY	FEE	DOB	GPA
1	Unzila	3rd year	Meerut	110188	01-JUN-02	-
3	Ananya	1st year	Noida	125000	21-JAN-04	-
4	Kamlesh	3rd year	Dehradun	120100	29-DEC-02	-
5	Samya	3rd year	Karnal	120100	04-AUG-02	-
6	Imad	2nd year	Meerut	125000	21-DEC-03	-
7	Anant	4th year	Banglore	145000	11-SEP-00	-
8	Tanishka	4th year	Rishikesh	150000	05-APR-01	-
9	Vivek	3rd year	Srinagar	135000	14-DEC-01	-

Download CSV
8 rows selected.

- Display the following output:
Sample output (in a single column, column name 'Result'):
Suraj having a candidate id 101 who studies in class IX and lives in Dehradun has a GPA of 7.

SQL Worksheet   Clear   Find   Actions ∨   Save   Run ▶

```
1  select candidate_name || ' having a candidate id ' || candidate_id || ' who is in ' || class || ' and lives in ' || city
2  || ' has a GPA of ' || gpa as "Result" from candidate where candidate_name ='Vivek';
```

Result
Vivek having a candidate id 9 who is in 3rd year and lives in Srinagar has a GPA of 8.9

Download CSV

Database Schema for a Employee-Detail
employee(emp_id, emp_name, dateofbirth, designation, department, salary, gender)

- Create the tables with the appropriate integrity constraints.

SQL Worksheet   Clear   Find   Actions ∨   Save   Run ▶

```
1    create table Employee(
2    emp_id int PRIMARY KEY,
3    emp_name varchar2(20),
4    dateofbirth date,
5    designation varchar2(20),
6    department varchar2(20),
7    salary int,
8    gender varchar(10)
9    );
10
11   describe Employee;
```

TABLE EMPLOYEE

Column	Null?	Type
EMP_ID	NOT NULL	NUMBER
EMP_NAME	-	VARCHAR2(20)
DATEOFBIRTH	-	DATE
DESIGNATION	-	VARCHAR2(20)
DEPARTMENT	-	VARCHAR2(20)
SALARY	-	NUMBER
GENDER	-	VARCHAR2(10)

- Insert 5 records in each of the tables

SQL Worksheet   Clear   Find   Actions ∨   Save   Run ▶

```
1    insert into Employee values(101, 'Unzila', '1-JUN-95', 'Manager', 'Development', 100000, 'Female');
2    insert into Employee values(102, 'Samya', '21-AUG-92', 'Chief Architect', 'Architecture', 80000, 'Female');
3    insert into Employee values(103, 'Vivek', '15-FEB-85', 'Security Officer', 'Security', 175000, 'Male');
4    insert into Employee values(104, 'Nitin', '18-DEC-91', 'Product Manager', 'Marketing', 140000, 'Male');
5    insert into Employee values(105, 'Kamlesh', '16-NOV-89', 'Software Engineer', 'IT', 130000, 'Male');
6    select * from Employee;
```

EMP_ID	EMP_NAME	DATEOFBIRTH	DESIGNATION	DEPARTMENT	SALARY	GENDER
105	Kamlesh	16-NOV-89	Software Engineer	IT	130000	Male
101	Unzila	01-JUN-95	Manager	Development	100000	Female
102	Samya	21-AUG-92	Chief Architect	Architecture	80000	Female
104	Nitin	18-DEC-91	Product Manager	Marketing	140000	Male
103	Vivek	15-FEB-85	Security Officer	Security	175000	Male

Download CSV
5 rows selected.

- List the employee details whose name does not contain 'b'.

SQL Worksheet   Clear   Find   Actions ∨   Save   Run ▶

```
1    select * from Employee where emp_name not like 'b%b%b'
```

EMP_ID	EMP_NAME	DATEOFBIRTH	DESIGNATION	DEPARTMENT	SALARY	GENDER
105	Kamlesh	16-NOV-89	Software Engineer	IT	130000	Male
101	Unzila	01-JUN-95	Manager	Development	100000	Female
102	Samya	21-AUG-92	Chief Architect	Architecture	80000	Female
104	Nitin	18-DEC-91	Product Manager	Marketing	140000	Male
103	Vivek	15-FEB-85	Security Officer	Security	175000	Male

Download CSV
5 rows selected.

- List the minimum, maximum, average salaries of employees.

SQL Worksheet Clear Find Actions ∨ Save Run ▶

```
1  select min(salary), max(salary), avg(salary) from Employee
```

MIN(SALARY)	MAX(SALARY)	AVG(SALARY)
80000	175000	125000

Download CSV

- Display the number of employees working in each department and their department name (Using group by).

SQL Worksheet Clear Find Actions ∨ Save Run ▶

```
1  select department, count(*) as "Number_of_Employees" from Employee group by department
2  |
```

DEPARTMENT	Number_of_Employees
Architecture	1
Marketing	1
Security	1
IT	1
Development	1

Download CSV
5 rows selected.

- List the Male programmers earning below the average salary of female programmers (Using group by).

SQL Worksheet Clear Find Actions ∨ Save Run ▶

```
1  select emp_name from Employee where gender='Male' and salary <
2  (select avg(salary) from Employee where gender='Female')
```

no data found

-Who is the youngest programmer born in 1995(Using Nested query).

SQL Worksheet Clear Find Actions ∨ Save Run ▶

```
1  select emp_name from Employee where dateofbirth=
2  (select max(dateofbirth) from Employee where to_char(dateofbirth, 'YYYY') like '1995')
3
```

EMP_NAME
Unzila

Download CSV

Database Schema for a customer-sale scenario
Customer(Cust id : integer, cust_name: string)
Item(item_id: integer, item_name: string, price: integer)
Sale(bill_no: integer, bill_date: date, cust_id: integer, item_id: integer, qty_sold: integer)

- Create the tables with the appropriate integrity constraints

SQL Worksheet   Clear   Find   Actions ∨   Save   Run ▶

```
1   create table Customer(
2   cust_id int PRIMARY KEY,
3   cust_name varchar2(20)
4   )
5
6   describe Customer;
```

TABLE CUSTOMER

Column	Null?	Type
CUST_ID	NOT NULL	NUMBER
CUST_NAME	-	VARCHAR2(20)

Download CSV
2 rows selected.

SQL Worksheet   Clear   Find   Actions ∨   Save   Run ▶

```
1   create table Sale(
2   bill_no int PRIMARY KEY,
3   bill_date date,
4   cust_id int,
5   item_id int,
6   qty_sold int
7   );
8
9   describe Sale;
```

TABLE SALE

Column	Null?	Type
BILL_NO	NOT NULL	NUMBER
BILL_DATE	-	DATE
CUST_ID	-	NUMBER
ITEM_ID	-	NUMBER
QTY_SOLD	-	NUMBER

Download CSV
5 rows selected.

SQL Worksheet   Clear   Find   Actions ∨   Save   Run ▶

```
1   create table Item(
2   item_id int PRIMARY KEY,
3   item_name varchar2(20),
4   price int
5   );
6
7   describe Item;
```

TABLE ITEM

Column	Null?	Type
ITEM_ID	NOT NULL	NUMBER
ITEM_NAME	-	VARCHAR2(20)
PRICE	-	NUMBER

Download CSV
3 rows selected.

- Insert around 5-10 records in each of the tables

SQL Worksheet   Clear   Find   Actions ∨   Save   Run ▶

```
1   insert into Customer values(1, 'Unzila');
2   insert into Customer values(2, 'Vivek');
3   insert into Customer values(3, 'Kamlesh');
4   insert into Customer values(4, 'Nitin');
5   insert into Customer values(5, 'Samya');
6
7   select * from Customer;
```

CUST_ID	CUST_NAME
1	Unzila
2	Vivek
3	Kamlesh
4	Nitin
5	Samya

Download CSV
5 rows selected.

```
1  insert into Item values(101, 'Watch', 1000);
2  insert into Item values(102, 'Tshirts', 800);
3  insert into Item values(103, 'Shoes', 2000);
4  insert into Item values(104, 'Trousers', 1500);
5  insert into Item values(105, 'Hoodies', 1200);
6
7  Select * from Item;
```

ITEM_ID	ITEM_NAME	PRICE
101	Watch	1000
102	Tshirts	800
103	Shoes	2000
104	Trousers	1500
105	Hoodies	1200

Download CSV
5 rows selected.

```
1   insert into Sale values(210, '27-Nov-2022', 1, 101, 2);
2   insert into Sale values(211, '19-Jun-2022', 2, 102, 5);
3   insert into Sale values(212, '5-Nov-2022', 1, 103, 2);
4   insert into Sale values(213, '27-Nov-2022', 4, 104, 3);
5   insert into Sale values(214, '9-Feb-2022', 5, 105, 6);
6   insert into Sale values(215, '27-Nov-2022', 3, 103, 2);
7   insert into Sale values(216, '13-Jun-2022', 2, 103, 1);
8   insert into Sale values(217, '27-Nov-2022', 4, 104, 2);
9   insert into Sale values(218, '21-Nov-2022', 3, 101, 3);
10  insert into Sale values(219, '27-Nov-2022', 1, 102, 8);
11  select * from Sale;
```

BILL_NO	BILL_DATE	CUST_ID	ITEM_ID	QTY_SOLD
210	27-NOV-22	1	101	2
211	19-JUN-22	2	102	5
212	05-NOV-22	1	103	2
213	27-NOV-22	4	104	3
214	09-FEB-22	5	105	6
215	27-NOV-22	3	103	2
216	13-JUN-22	2	103	1
217	27-NOV-22	4	104	2
218	21-NOV-22	3	101	3
219	27-NOV-22	1	102	8

Download CSV
10 rows selected.

- List all the bills for the current date with the customer names and item numbers.

```
1  select Customer.cust_name, Item.item_id, Sale.bill_no from Customer, Item, Sale where Customer.cust_id=Sale.cust_id and Item.item_id=Sale.item_id and Sale.bill_date=to_char(sysdate);
2
```

CUST_NAME	ITEM_ID	BILL_NO
Unzila	101	210
Nitin	104	213
Kamlesh	103	215
Nitin	104	217
Unzila	102	219

Download CSV
5 rows selected.

- List the total Bill details with the quantity sold, price of the item and the final amount

```
1  select Customer.cust_name, Item.item_name, Item.price, (Item.price*Sale.qty_sold) as Final_Amount from Customer, Item, Sale
2  where Customer.cust_id=Sale.cust_id and Item.item_id=Sale.item_id and Sale.bill_date='19-Jun-22'
```

CUST_NAME	ITEM_NAME	PRICE	FINAL_AMOUNT
Vivek	Tshirts	800	4000

Download CSV

- List the details of the customer who have bought a product which has a price>1500

```
1  select Customer.cust_id, Customer.cust_name from Customer, Item, Sale
2  where Customer.cust_id=Sale.cust_id and Item.item_id=Sale.item_id and price>1500
```

CUST_ID	CUST_NAME
1	Unzila
2	Vivek
3	Kamlesh

Download CSV
3 rows selected.

- Give a count of how many products have been bought by each customer.

```
1  select cust_id, count(item_id) as Total_Products from Sale group by Sale.cust_id
```

CUST_ID	TOTAL_PRODUCTS
1	3
2	2
4	2
5	1
3	2

Download CSV
5 rows selected.

- Give a list of products bought by a customer having cust_id as 5

```
1  select Item.item_name from Customer, Item, Sale where Customer.cust_id=Sale.cust_id and Item.item_id=Sale.item_id and Sale.cust_id=5
```

ITEM_NAME
Hoodies

Download CSV

- List the item details which are sold as of today.

```
1  select Item.item_id, Item.item_name, Item.price from Item, Sale
2  where Item.item_id=Sale.item_id
3  and Sale.bill_date=to_char(sysdate);
```

ITEM_ID	ITEM_NAME	PRICE
101	Watch	1000
104	Trousers	1500
103	Shoes	2000
104	Trousers	1500
102	Tshirts	800

Download CSV
5 rows selected.

Database Schema for a Student Library scenario
Student(Stud_no : integer, Stud_name: string)
Membership(Mem_no: integer, Stud_no: integer)
Book(book_no: integer, book_name:string, author: string)
Iss_rec(iss_no:integer, iss_date: date, Mem_no: integer, book_no: integer)

- Create the tables with the appropriate integrity constraints

SQL Worksheet   Clear   Find   Actions ▾   Save   Run ▶

```
1  create table Student(
2  Stud_no int PRIMARY KEY,
3  Stud_name varchar2(20)
4  );
5
6  describe Student;
```

TABLE STUDENT

Column	Null?	Type
STUD_NO	NOT NULL	NUMBER
STUD_NAME	-	VARCHAR2(20)

Download CSV
2 rows selected.

SQL Worksheet   Clear   Find   Actions ▾   Save   Run ▶

```
1  create table Membership(
2  Mem_no int PRIMARY KEY,
3  Stud_no int
4  );
5  
6  describe Membership;
```

TABLE MEMBERSHIP

Column	Null?	Type
MEM_NO	NOT NULL	NUMBER
STUD_NO	-	NUMBER

Download CSV
2 rows selected.

SQL Worksheet   Clear   Find   Actions ▾   Save   Run ▶

```
1  create table Book(
2  Book_no int PRIMARY KEY,
3  Book_name varchar2(40),
4  Author varchar2(40)
5  )
6
7  describe Book;
```

TABLE BOOK

Column	Null?	Type
BOOK_NO	NOT NULL	NUMBER
BOOK_NAME	-	VARCHAR2(40)
AUTHOR	-	VARCHAR2(40)

Download CSV
3 rows selected.

SQL Worksheet

```
1  create table Iss_rec(
2  iss_no int PRIMARY KEY,
3  iss_date date,
4  mem_no int,
5  book_no int
6  );
7  describe Iss_rec;
```

TABLE ISS_REC

Column	Null?	Type
ISS_NO	NOT NULL	NUMBER
ISS_DATE	-	DATE
MEM_NO	-	NUMBER
BOOK_NO	-	NUMBER

Download CSV
4 rows selected.

- Insert around 5-10 records in each of the tables.

SQL Worksheet

```
1  insert into Student values(1, 'Unzila');
2  insert into Student values(2, 'Vivek');
3  insert into Student values(3, 'Samya');
4  insert into Student values(4, 'Kamlesh');
5  insert into Student values(5, 'Priyanshu');
6  Select * from Student;
7
```

STUD_NO	STUD_NAME
1	Unzila
2	Vivek
3	Samya
4	Kamlesh
5	Priyanshu

Download CSV
5 rows selected.

SQL Worksheet

```
1  insert into Membership values(101, 1);
2  insert into Membership values(102, 2);
3  insert into Membership values(103, 3);
4  insert into Membership values(104, 4);
5  insert into Membership values(105, 5);
6  select * from Membership;
7
```

MEM_NO	STUD_NO
101	1
102	2
103	3
104	4
105	5

Download CSV
5 rows selected.

```
1   insert into Book values(1000, 'C:The Complete Reference', 'Herbert Schildt');
2   insert into Book values(2000, 'Head First Java', 'Kathy Sierra');
3   insert into Book values(3000, 'Introduction to Algorithm', 'Ronald Rivest');
4   insert into Book values(4000, 'Database Management System', 'Raghu Ramakrishnan');
5   insert into Book values(5000, 'Operating System', 'Avi Silberschatz');
6   select * from Book;
```

BOOK_NO	BOOK_NAME	AUTHOR
1000	C:The Complete Reference	Herbert Schildt
2000	Head First Java	Kathy Sierra
3000	Introduction to Algorithm	Ronald Rivest
4000	Database Management System	Raghu Ramakrishnan
5000	Operating System	Avi Silberschatz

Download CSV
5 rows selected.

```
1    insert into Iss_rec values(200, '27-Nov-2022', 101, 4000);
2    insert into Iss_rec values(210, '21-Nov-2022', 104, 2000);
3    insert into Iss_rec values(220, '25-Nov-2022', 102, 1000);
4    insert into Iss_rec values(230, '02-Nov-2022', 101, 3000);
5    insert into Iss_rec values(240, '27-Nov-2022', 103, 5000);
6    insert into Iss_rec values(250, '27-Nov-2022', 105, 3000);
7    insert into Iss_rec values(260, '15-Nov-2022', 102, 5000);
8    insert into Iss_rec values(270, '27-Nov-2022', 103, 2000);
9    insert into Iss_rec values(280, '18-Nov-2022', 102, 4000);
10   insert into Iss_rec values(290, '27-Nov-2022', 101, 2000);
11   Select * from Iss_rec;
```

ISS_NO	ISS_DATE	MEM_NO	BOOK_NO
200	27-NOV-22	101	4000
210	21-NOV-22	104	2000
220	25-NOV-22	102	1000
230	02-NOV-22	101	3000
240	27-NOV-22	103	5000
250	27-NOV-22	105	3000
260	15-NOV-22	102	5000
270	27-NOV-22	103	2000
280	18-NOV-22	102	4000
290	27-NOV-22	101	2000

Download CSV
10 rows selected.

- List all the student names with their membership numbers.

```
1   Select Student.stud_name, Membership.mem_no from Student, Membership where Student.stud_no=Membership.stud_no
```

STUD_NAME	MEM_NO
Unzila	101
Vivek	102
Samya	103
Kamlesh	104
Priyanshu	105

Download CSV
5 rows selected.

- List all the issues for the current date with student and Book names.

SQL Worksheet   Clear   Find   Actions ∨   Save   **Run ▶**

```
1  Select Iss_rec.iss_no, Student.stud_name, Book.book_name from Student, Membership, Book, Iss_rec
2  where Student.stud_no=Membership.stud_no and Iss_rec.mem_no=Membership.mem_no and Book.Book_no=Iss_rec.book_no
3  and Iss_rec.iss_date=to_char(sysdate)
```

ISS_NO	STUD_NAME	BOOK_NAME
200	Unzila	Database Management System
240	Samya	Operating System
250	Priyanshu	Introduction to Algorithm
270	Samya	Head First Java
290	Unzila	Head First Java

Download CSV
5 rows selected.

- List the details of students who borrowed book whose author is Raghu Ramakrishnan.

SQL Worksheet   Clear   Find   Actions ∨   Save   **Run ▶**

```
1  Select Student.stud_no, Student.stud_name from Student, Membership, Book, Iss_rec
2  where Student.stud_no=Membership.stud_no and Iss_rec.mem_no=Membership.mem_no and Book.Book_no=Iss_rec.book_no
3  and Book.author='Raghu Ramakrishnan'
```

STUD_NO	STUD_NAME
1	Unzila
2	Vivek

Download CSV
2 rows selected.

- Give a count of how many books have been bought by each student.

SQL Worksheet   Clear   Find   Actions ∨   Save   **Run ▶**

```
1  Select Student.stud_name, count(Iss_rec.book_no) as No_of_Books from Student, Membership, Book, Iss_rec
2  where Student.stud_no=Membership.stud_no and Iss_rec.mem_no=Membership.mem_no and Book.book_no=Iss_rec.book_no
3  group by Student.stud_name;
```

STUD_NAME	NO_OF_BOOKS
Priyanshu	1
Kamlesh	1
Unzila	3
Samya	2
Vivek	3

Download CSV
5 rows selected.

- Give a list of books taken by student with stud_no as 2

SQL Worksheet   Clear   Find   Actions ∨   Save   **Run ▶**

```
1  Select Book.Book_name from Student, Membership, Book, Iss_rec
2  where Student.stud_no=Membership.stud_no and Iss_rec.mem_no=Membership.mem_no and Book.book_no=Iss_rec.book_no
3  and Student.stud_no=2;
```

BOOK_NAME
C:The Complete Reference
Operating System
Database Management System

Download CSV
3 rows selected.

- List the book details which are issued as of today.

```
1   Select Book.Book_no, Book.Book_name, Book.author from Book, Iss_rec
2   where Book.book_no=Iss_rec.book_no and Iss_rec.iss_date=to_char(sysdate);
```

BOOK_NO	BOOK_NAME	AUTHOR
4000	Database Management System	Raghu Ramakrishnan
5000	Operating System	Avi Silberschatz
3000	Introduction to Algorithm	Ronald Rivest
2000	Head First Java	Kathy Sierra
2000	Head First Java	Kathy Sierra

Download CSV
5 rows selected.

SQL Worksheet

◇ Clear   ⚡ Find   Actions ⌄   💾 Save   Run ▶

```
1   Select Book.Book_no, Book.Book_name, Book.author from Book, Iss_rec
2   where Book.book_no=Iss_rec.book_no and Iss_rec.iss_date=to_char(sysdate);
```

BOOK_NO	BOOK_NAME	AUTHOR
4000	Database Management System	Raghu Ramakrishnan
5000	Operating System	Avi Silberschatz

Database Schema for a Employee-pay scenario:
employee(emp_id : integer, emp_name: string)
department(dept_id: integer, dept_name:string)
paydetails(emp_id : integer, dept_id: integer, basic: integer, deductions: integer, additions: integer, DOJ: date)
payroll(emp_id : integer, pay_date: date)

- Create the tables with the appropriate integrity constraints

SQL Worksheet   Clear   Find   Actions ∨   Save   Run ▶

```
1  create table Employee(
2  emp_id int PRIMARY KEY,
3  emp_name varchar2(20)
4  )
5
6  describe Employee
7
```

TABLE EMPLOYEE

Column	Null?	Type
EMP_ID	NOT NULL	NUMBER
EMP_NAME	-	VARCHAR2(20)

Download CSV
2 rows selected.

SQL Worksheet   Clear   Find   Actions ∨   Save   Run ▶

```
1  create table Department(
2  dept_id int PRIMARY KEY,
3  dept_name varchar2(20)
4  )
5
6  describe Department
7
```

TABLE DEPARTMENT

Column	Null?	Type
DEPT_ID	NOT NULL	NUMBER
DEPT_NAME	-	VARCHAR2(20)

Download CSV
2 rows selected.

SQL Worksheet   Clear   Find   Actions ∨   Save   Run ▶

```
1  create table PayDetails( emp_id int, dept_id int, basic int, deductions int, additions int, DOJ date )
2  describe PayDetails
3  |
```

TABLE PAYDETAILS

Column	Null?	Type
EMP_ID	-	NUMBER
DEPT_ID	-	NUMBER
BASIC	-	NUMBER
DEDUCTIONS	-	NUMBER
ADDITIONS	-	NUMBER
DOJ	-	DATE

Download CSV
6 rows selected.

```
1   create table Payroll( emp_id int, pay_date date )
2   describe Payroll;
3   |
```

TABLE PAYROLL

Column	Null?	Type
EMP_ID	-	NUMBER
PAY_DATE	-	DATE

Download CSV
2 rows selected.

- Insert around 10 records in each of the tables.

```
1    insert into Employee values(100, 'Nitin');
2    insert into Employee values(101, 'Arinjay');
3    insert into Employee values(102, 'Vivek');
4    insert into Employee values(103, 'Priyanshu');
5    insert into Employee values(104, 'Kamlesh');
6    insert into Employee values(105, 'Adesh');
7    insert into Employee values(106, 'Abhinav');
8    insert into Employee values(107, 'Junaid');
9    insert into Employee values(108, 'Karan');
10   insert into Employee values(109, 'Siddharth');
11   Select * from Employee;|
```

EMP_ID	EMP_NAME
100	Nitin
101	Arinjay
102	Vivek
103	Priyanshu
104	Kamlesh
105	Adesh
106	Abhinav
107	Junaid
108	Karan
109	Siddharth

Download CSV
10 rows selected.

```
1   insert into Department values(1, 'Development');
2   insert into Department values(2, 'Sales & Marketing');
3   insert into Department values(3, 'Architecture');
4   insert into Department values(4, 'Security');
5   insert into Department values(5, 'Research');
6   Select * from Department;
7   |
```

DEPT_ID	DEPT_NAME
1	Development
2	Sales & Marketing
3	Architecture
4	Security
5	Research

Download CSV
5 rows selected.

```
1   insert into Payroll values(100, '15-Nov-22');
2   insert into Payroll values(101, '10-Nov-22');
3   insert into Payroll values(102, '20-Oct-22');
4   insert into Payroll values(103, '01-Aug-22');
5   insert into Payroll values(104, '15-Sep-22');
6   insert into Payroll values(105, '30-Sep-22');
7   insert into Payroll values(106, '25-Nov-22');
8   insert into Payroll values(107, '20-Nov-22');
9   insert into Payroll values(108, '15-Oct-22');
10  insert into Payroll values(109, '29-Aug-22');
11  Select * from Payroll
```

EMP_ID	PAY_DATE
100	15-NOV-22
101	10-NOV-22
102	20-OCT-22
103	01-AUG-22
104	15-SEP-22
105	30-SEP-22
106	25-NOV-22
107	20-NOV-22
108	15-OCT-22
109	29-AUG-22

Download CSV
10 rows selected.

```
1   insert into PayDetails values(100, 2, 80000, 2000, 5000, '8-Jan-2015');
2   insert into PayDetails values(101, 1, 100000, 1500, 3500, '24-Mar-2002');
3   insert into PayDetails values(102, 3, 90000, 1000, 4000, '28-Nov-2008');
4   insert into PayDetails values(103, 2, 120000, 500, 2000, '15-Feb-2005');
5   insert into PayDetails values(104, 1, 95000, 200, 1000, '1-Aug-2018');
6   insert into PayDetails values(105, 4, 100000, 3000, 5000, '12-Jul-2015');
7   insert into PayDetails values(106, 5, 50000, 100, 1000, '18-Jun-2012');
8   insert into PayDetails values(107, 5, 60000, 500, 700, '20-Nov-2021');
9   insert into PayDetails values(108, 3, 75000, 1000, 1500, '25-Sep-2019');
10  insert into PayDetails values(109, 4, 110000, 600, 2500, '1-Mar-2010');
11  Select * from PayDetails
```

EMP_ID	DEPT_ID	BASIC	DEDUCTIONS	ADDITIONS	DOJ
101	1	100000	1500	3500	24-MAR-02
102	3	90000	1000	4000	28-NOV-08
103	2	120000	500	2000	15-FEB-05
104	1	95000	200	1000	01-AUG-18
105	4	100000	3000	5000	12-JUL-15
106	5	50000	100	1000	18-JUN-12
107	5	60000	500	700	20-NOV-21
108	3	75000	1000	1500	25-SEP-19
109	4	110000	600	2500	01-MAR-10
100	2	80000	2000	5000	08-JAN-15

Download CSV
10 rows selected.

- List the employee details department wise.

```
1   Select Employee.emp_id, Employee.emp_name from Employee, PayDetails where Employee.emp_id=PayDetails.emp_id
2
```

EMP_ID	EMP_NAME
101	Arinjay
102	Vivek
103	Priyanshu
104	Kamlesh
105	Adesh
106	Abhinav
107	Junaid
108	Karan
109	Siddharth
100	Nitin

Download CSV
10 rows selected.

- List all the employee names who joined after particular date.

SQL Worksheet ⬦ Clear ✎ Find Actions ⌄ 💾 Save Run ▶

```
1  Select Employee.emp_name from Employee, PayDetails
2  where Employee.emp_id=PayDetails.emp_id and doj>'31-Dec-2012'
```

EMP_NAME
Nitin
Kamlesh
Adesh
Junaid
Karan

Download CSV
5 rows selected.

- List the details of employees whose basic salary is between 1,00,000 and 2,00,000.

SQL Worksheet ⬦ Clear ✎ Find Actions ⌄ 💾 Save Run ▶

```
1  Select Employee.emp_id, Employee.emp_name from Employee, PayDetails
2  where Employee.emp_id=PayDetails.emp_id and PayDetails.basic between 100000 and 200000
```

EMP_ID	EMP_NAME
101	Arinjay
103	Priyanshu
105	Adesh
109	Siddharth

Download CSV
4 rows selected.

- Give a count of how many employees are working in each department.

SQL Worksheet ⬦ Clear ✎ Find Actions ⌄ 💾 Save Run ▶

```
1  Select PayDetails.dept_id, count(PayDetails.emp_id) as No_of_Employees from Department, PayDetails
2  where Department.dept_id=PayDetails.dept_id group by PayDetails.dept_id
```

DEPT_ID	NO_OF_EMPLOYEES
1	2
2	2
4	2
5	2
3	2

Download CSV
5 rows selected.

- Give a names of the employees whose netsalary>1,00,000.

SQL Worksheet ⬦ Clear ✎ Find Actions ⌄ 💾 Save Run ▶

```
1  Select emp_name from Employee
2  where emp_id in(Select emp_id from PayDetails where(Basic-Deductions>100000))
```

EMP_NAME
Priyanshu
Siddharth

Download CSV
2 rows selected.

- List the details for an employee_id=5

```
1  Select * from Employee where emp_id=105;
```

EMP_ID	EMP_NAME
105	Adesh

Download CSV

Database Schema for a Video Library scenario:
Customer(cust_no: integer,cust_name: string)
Membership(Mem_no: integer, cust_no: integer)
Cassette(cass_no:integer, cass_name:string, Language: String)
Iss_rec(iss_no: integer, iss_date: date, mem_no: integer, cass_no: integer)

- Create the tables with the appropriate integrity constraints.

SQL Worksheet   ◇ Clear   ✎ Find   Actions ∨   💾 Save   Run ▶

```
1  create table Customer(
2  cust_no int,
3  cust_name varchar2(20)
4  )
5  Describe Customer
```

TABLE CUSTOMER

Column	Null?	Type
CUST_NO	-	NUMBER
CUST_NAME	-	VARCHAR2(20)

Download CSV
2 rows selected.

SQL Worksheet   ◇ Clear   ✎ Find   Actions ∨   💾 Save   Run ▶

```
1  create table Membership(
2  mem_no int PRIMARY KEY,
3  cust_no int
4  )
5  Describe Membership
```

TABLE MEMBERSHIP

Column	Null?	Type
MEM_NO	NOT NULL	NUMBER
CUST_NO	-	NUMBER

Download CSV
2 rows selected.

SQL Worksheet   ◇ Clear   ✎ Find   Actions ∨   💾 Save   Run ▶

```
1  create table Cassette(
2  cass_no int PRIMARY KEY,
3  cass_name varchar2(40),
4  language varchar2(20)
5  );
6  Describe Cassette
```

TABLE CASSETTE

Column	Null?	Type
CASS_NO	NOT NULL	NUMBER
CASS_NAME	-	VARCHAR2(40)
LANGUAGE	-	VARCHAR2(20)

Download CSV
3 rows selected.

SQL Worksheet

```
1   create table Iss_rec( iss_no int PRIMARY KEY, iss_date date, mem_no int, cass_no int )
2   Describe Iss_rec;
```

TABLE ISS_REC

Column	Null?	Type
ISS_NO	NOT NULL	NUMBER
ISS_DATE	-	DATE
MEM_NO	-	NUMBER
CASS_NO	-	NUMBER

Download CSV
4 rows selected.

SQL Worksheet

```
1   insert into Customer values(1, 'Ananya');
2   insert into Customer values(2, 'Vivek');
3   insert into Customer values(3, 'Samya');
4   insert into Customer values(4, 'Harsh');
5   insert into Customer values(5, 'Aryan');
6   Select * from Customer
7
```

CUST_NO	CUST_NAME
1	Ananya
2	Vivek
3	Samya
4	Harsh
5	Aryan

Download CSV
5 rows selected.

- Insert around 10 records in each of the tables.

SQL Worksheet

```
1   insert into Membership values(101, 1);
2   insert into Membership values(102, 2);
3   insert into Membership values(103, 3);
4   insert into Membership values(104, 4);
5   insert into Membership values(105, 5);
6   Select * from Membership;
```

MEM_NO	CUST_NO
101	1
102	2
103	3
104	4
105	5

Download CSV
5 rows selected.

```
1  insert into Cassette values(1000, 'Veer Zara', 'Hindi');
2  insert into Cassette values(2000, 'Mr President', 'English');
3  insert into Cassette values(3000, 'Ejaman', 'Tamil');
4  insert into Cassette values(4000, 'Yathramozhi', 'Malyalam');
5  insert into Cassette values(5000, 'Akhiyaan Udeek Diyan', 'Punjabi');
6  Select * from Cassette;
```

CASS_NO	CASS_NAME	LANGUAGE
1000	Veer Zara	Hindi
2000	Mr President	English
3000	Ejaman	Tamil
4000	Yathramozhi	Malyalam
5000	Akhiyaan Udeek Diyan	Punjabi

Download CSV
5 rows selected.

```
1   insert into Iss_rec values(200, '27-Nov-2022', 101, 4000);
2   insert into Iss_rec values(210, '21-Nov-2022', 104, 2000);
3   insert into Iss_rec values(220, '25-Nov-2022', 102, 1000);
4   insert into Iss_rec values(230, '02-Nov-2022', 101, 3000);
5   insert into Iss_rec values(240, '27-Nov-2022', 103, 5000);
6   insert into Iss_rec values(250, '27-Nov-2022', 105, 3000);
7   insert into Iss_rec values(260, '15-Nov-2022', 102, 5000);
8   insert into Iss_rec values(270, '27-Nov-2022', 103, 2000);
9   insert into Iss_rec values(280, '18-Nov-2022', 102, 4000);
10  insert into Iss_rec values(290, '27-Nov-2022', 101, 2000);
11  Select * from Iss_rec;
```

ISS_NO	ISS_DATE	MEM_NO	CASS_NO
200	27-NOV-22	101	4000
210	21-NOV-22	104	2000
220	25-NOV-22	102	1000
230	02-NOV-22	101	3000
240	27-NOV-22	103	5000
250	27-NOV-22	105	3000
260	15-NOV-22	102	5000
270	27-NOV-22	103	2000
280	18-NOV-22	102	4000
290	27-NOV-22	101	2000

Download CSV
10 rows selected.

- List all the customer names with their membership numbers.

```
1  Select Customer.cust_name, Membership.mem_no from Customer, Membership
2  where Customer.cust_no=Membership.cust_no
```

CUST_NAME	MEM_NO
Ananya	101
Vivek	102
Samya	103
Harsh	104
Aryan	105

Download CSV
5 rows selected.

- List all the issues for the current date with the customer names and cassette names.

SQL Worksheet   Clear   Find   Actions ∨   Save   Run ▶

```
1  Select Iss_rec.iss_no, Customer.cust_name, Cassette.cass_name from Customer, Membership, Cassette, Iss_rec
2  where Customer.cust_no=Membership.cust_no and Iss_rec.mem_no=Membership.mem_no and Cassette.cass_no=Iss_rec.cass_no
3  and Iss_rec.iss_date=to_char(sysdate);
```

ISS_NO	CUST_NAME	CASS_NAME
270	Samya	Mr President
290	Ananya	Mr President
250	Aryan	Ejaman
200	Ananya	Yathramozhi
240	Samya	Akhiyaan Udeek Diyan

Download CSV
5 rows selected.

- List the details of the customer who has borrowed the cassette whose title is "Veer Zara".

SQL Worksheet   Clear   Find   Actions ∨   Save   Run ▶

```
1  Select Customer.cust_no, Customer.cust_name from Customer, Membership, Cassette, Iss_rec
2  where Customer.cust_no=Membership.cust_no and Iss_rec.mem_no=Membership.mem_no and Cassette.cass_no=Iss_rec.cass_no
3  and Cassette.cass_name='Veer Zara'
```

CUST_NO	CUST_NAME
2	Vivek

Download CSV

- Give a count of how many cassettes have been borrowed by each customer.

SQL Worksheet   Clear   Find   Actions ∨   Save   Run ▶

```
1  Select Customer.cust_name, count(Iss_rec.cass_no) as No_of_Cassettes from Customer, Membership, Cassette, Iss_rec
2  where Customer.cust_no=Membership.cust_no and Iss_rec.mem_no=Membership.mem_no and Cassette.cass_no=Iss_rec.cass_no
3  group by Customer.cust_name
4
```

CUST_NAME	NO_OF_CASSETTES
Harsh	1
Samya	2
Vivek	3
Aryan	1
Ananya	3

Download CSV
5 rows selected.

- Give a list of book which has been taken by the student with mem_no as 5.

SQL Worksheet   Clear   Find   Actions ∨   Save   Run ▶

```
1  Select Cassette.cass_name from Customer, Membership, Cassette, Iss_rec
2  where Customer.cust_no=Membership.cust_no and Iss_rec.mem_no=Membership.mem_no and Cassette.cass_no=Iss_rec.cass_no
3  and Customer.cust_no=5
4
```

CASS_NAME
Ejaman

Download CSV

- List the cassettes issues for today

SQL Worksheet

```
1  Select Cassette.cass_name from Cassette, Iss_rec
2  where Cassette.cass_no=Iss_rec.cass_no and Iss_rec.iss_date=to_char(sysdate)
3
```

CASS_NAME
Yathramozhi
Akhiyaan Udeek Diyan
Ejaman
Mr President
Mr President

Download CSV
5 rows selected.

- Create a view which lists outs the iss_no, iss_date, cust_name, cass_name.

SQL Worksheet

```
1  Create view v1 as Select Iss_rec.iss_no, Iss_rec.iss_date, Customer.cust_name, Cassette.cass_name
2  from Customer, Membership, Cassette, Iss_rec
3  where Customer.cust_no=Membership.cust_no and Iss_rec.mem_no=Membership.mem_no and Cassette.cass_no=Iss_rec.cass_no;
4  Select * from v1;
5
```

ISS_NO	ISS_DATE	CUST_NAME	CASS_NAME
220	25-NOV-22	Vivek	Veer Zara
210	21-NOV-22	Harsh	Mr President
270	27-NOV-22	Samya	Mr President
290	27-NOV-22	Ananya	Mr President
230	02-NOV-22	Ananya	Ejaman
250	27-NOV-22	Aryan	Ejaman
200	27-NOV-22	Ananya	Yathramozhi
280	18-NOV-22	Vivek	Yathramozhi
240	27-NOV-22	Samya	Akhiyaan Udeek Diyan
260	15-NOV-22	Vivek	Akhiyaan Udeek Diyan

Download CSV
10 rows selected.

- Create a view which lists issues-date wise for the last one week.

SQL Worksheet

```
1  Create view v2 as Select iss_date from Iss_rec where iss_date>=(sysdate-7) order by iss_date;
2  Select * from v2;
3
```

ISS_DATE
25-NOV-22
27-NOV-22
27-NOV-22
27-NOV-22
27-NOV-22
27-NOV-22

Download CSV
6 rows selected.

# Database Schema for a student-Lab scenario:
Student(stud_no: integer, stud_name: string, class: string)
Class(class: string, descrip: string)
Lab(mach_no: integer, Lab_no: integer, description: String)
Allotment(Stud_no: Integer, mach_no: integer, dayof week: string)

- Create the tables with the appropriate integrity constraints.

SQL Worksheet   Clear   Find   Actions ⌄   Save   Run ▶

```
1   create table Student( stud_no int PRIMARY KEY, stud_name varchar2(20), class varchar2(10) );
2   Describe Student;
```

TABLE STUDENT

Column	Null?	Type
STUD_NO	NOT NULL	NUMBER
STUD_NAME	-	VARCHAR2(20)
CLASS	-	VARCHAR2(10)

Download CSV
3 rows selected.

SQL Worksheet   Clear   Find   Actions ⌄   Save   Run ▶

```
1   create table Class( class varchar2(20) PRIMARY KEY, descript varchar2(50) );
2   Describe Student;
```

TABLE STUDENT

Column	Null?	Type
STUD_NO	NOT NULL	NUMBER
STUD_NAME	-	VARCHAR2(20)
CLASS	-	VARCHAR2(10)

Download CSV
3 rows selected.

SQL Worksheet   Clear   Find   Actions ⌄   Save   Run ▶

```
1   create table Lab( mach_no int PRIMARY KEY, Lab_no int, Description varchar2(20) );
2   Describe Student;
```

TABLE STUDENT

Column	Null?	Type
STUD_NO	NOT NULL	NUMBER
STUD_NAME	-	VARCHAR2(20)
CLASS	-	VARCHAR2(10)

Download CSV
3 rows selected.

```
1   create table Allotment( stud_no int, mach_no int, DayOfWeek varchar2(20) );
2   Desc Allotment;
```

TABLE ALLOTMENT

Column	Null?	Type
STUD_NO	-	NUMBER
MACH_NO	-	NUMBER
DAYOFWEEK	-	VARCHAR2(20)

Download CSV

3 rows selected.

- Insert around 10 records in each of the tables.

```
1    insert into Student values(1, 'Unzila', 'CSE');
2    insert into Student values(2, 'Vivek', 'CSE');
3    insert into Student values(3, 'Samya', 'ECE');
4    insert into Student values(4, 'Nitin', 'CE');
5    insert into Student values(5, 'Priyanshu', 'ECE');
6    insert into Student values(6, 'Arinjay', 'ME');
7    insert into Student values(7, 'Kamlesh', 'CSE');
8    insert into Student values(8, 'Adesh', 'IT');
9    insert into Student values(9, 'Abhinav', 'ME');
10   insert into Student values(10, 'Tanishq', 'IT');
11   select * from Student
12
```

STUD_NO	STUD_NAME	CLASS
1	Unzila	CSE
2	Vivek	CSE
3	Samya	ECE
4	Nitin	CE
5	Priyanshu	ECE
6	Arinjay	ME
7	Kamlesh	CSE
8	Adesh	IT
9	Abhinav	ME
10	Tanishq	IT

Download CSV

10 rows selected.

```
1   insert into Class values('CSE', 'Computer Science Engineering');
2   insert into Class values('ECE', 'Electronics and Communication Engineering');
3   insert into Class values('CE', 'Chemical Engineering');
4   insert into Class values('ME', 'Mechanical Engineering');
5   insert into Class values('IT', 'Information Technology');
6   Select * from Class;
```

CLASS	DESCRIPT
CSE	Computer Science Engineering
ECE	Electronics and Communication Engineering
CE	Chemical Engineering
ME	Mechanical Engineering
IT	Information Technology

Download CSV

5 rows selected.

SQL Worksheet

```
1  insert into Lab values(1001, 101, 'Java Lab');
2  insert into Lab values(1002, 102, 'Electronics Lab');
3  insert into Lab values(1003, 103, 'Chemistry Lab');
4  insert into Lab values(1004, 104, 'Robotics Lab');
5  insert into Lab values(1005, 105, 'DSA Lab');
6  Select * from Lab;
```

MACH_NO	LAB_NO	DESCRIPTION
1001	101	Java Lab
1002	102	Electronics Lab
1003	103	Chemistry Lab
1004	104	Robotics Lab
1005	105	DSA Lab

Download CSV
5 rows selected.

SQL Worksheet

```
1   insert into Allotment values(1, 1005, 'Monday');
2   insert into Allotment values(2, 1005, 'Thursday');
3   insert into Allotment values(7, 1005, 'Tuesday');
4   insert into Allotment values(3, 1002, 'Wednesday');
5   insert into Allotment values(5, 1002, 'Wednesday');
6   insert into Allotment values(4, 1003, 'Saturday');
7   insert into Allotment values(6, 1004, 'Monday');
8   insert into Allotment values(9, 1004, 'Monday');
9   insert into Allotment values(8, 1001, 'Saturday');
10  insert into Allotment values(10, 1001, 'Saturday');
11  Select * from Allotment;
```

STUD_NO	MACH_NO	DAYOFWEEK
1	1005	Monday
2	1005	Thursday
7	1005	Tuesday
3	1002	Wednesday
5	1002	Wednesday
4	1003	Saturday
6	1004	Monday
9	1004	Monday
8	1001	Saturday
10	1001	Saturday

Download CSV
10 rows selected.

- **List all the machine allotments with the student names, lab and machine numbers.**

SQL Worksheet

```
1   Select Student.stud_name, Lab.lab_no, Lab.mach_no from Student, Lab, Allotment
2   where Student.stud_no=Allotment.stud_no and Lab.mach_no=Allotment.mach_no
```

STUD_NAME	LAB_NO	MACH_NO
Unzila	105	1005
Vivek	105	1005
Samya	102	1002
Nitin	103	1003
Priyanshu	102	1002
Arinjay	104	1004
Kamlesh	105	1005
Adesh	101	1001
Abhinav	104	1004
Tanishq	101	1001

Download CSV
10 rows selected.

- List the total number of lab allotments day wise.

```
1   Select DayOfWeek, count(mach_no) as No_of_Lab from Allotment group by DayOfWeek
```

DAYOFWEEK	NO_OF_LAB
Thursday	1
Tuesday	1
Saturday	3
Wednesday	2
Monday	3

Download CSV
5 rows selected.

- Give a count of how many machines have been allocated to the 'CSE' class.

```
1   Select count(mach_no) from Allotment where stud_no in(Select stud_no from Student where class='CSE')
```

COUNT(MACH_NO)
3

Download CSV

- Give a machine allotment details of the stud_no 5 with his personal and class details.

```
1   Select Allotment.stud_no, Allotment.mach_no, Student.stud_name, Student.class
2   from Student, Allotment where Allotment.stud_no=Student.stud_no and Student.stud_no=5
```

STUD_NO	MACH_NO	STUD_NAME	CLASS
5	1002	Priyanshu	ECE

Download CSV

- Count for how many machines have been allocated in Lab_no 1 for the day of the week as "Monday".

```
1   Select count(Allotment.mach_no) as No_of_Machines from Lab, Allotment
2   where Lab.mach_no=Allotment.mach_no and Lab.lab_no=101 and Allotment.dayofweek='Monday';
```

NO_OF_MACHINES
0

Download CSV

- How many students class wise have allocated machines in the labs.

```
1   Select class, count(stud_no) as No_of_Students from Student
2   where stud_no in (Select stud_no from Allotment)
3   group by class
```

CLASS	NO_OF_STUDENTS
CE	1
CSE	3
ECE	2
ME	2
IT	2

Download CSV
5 rows selected.