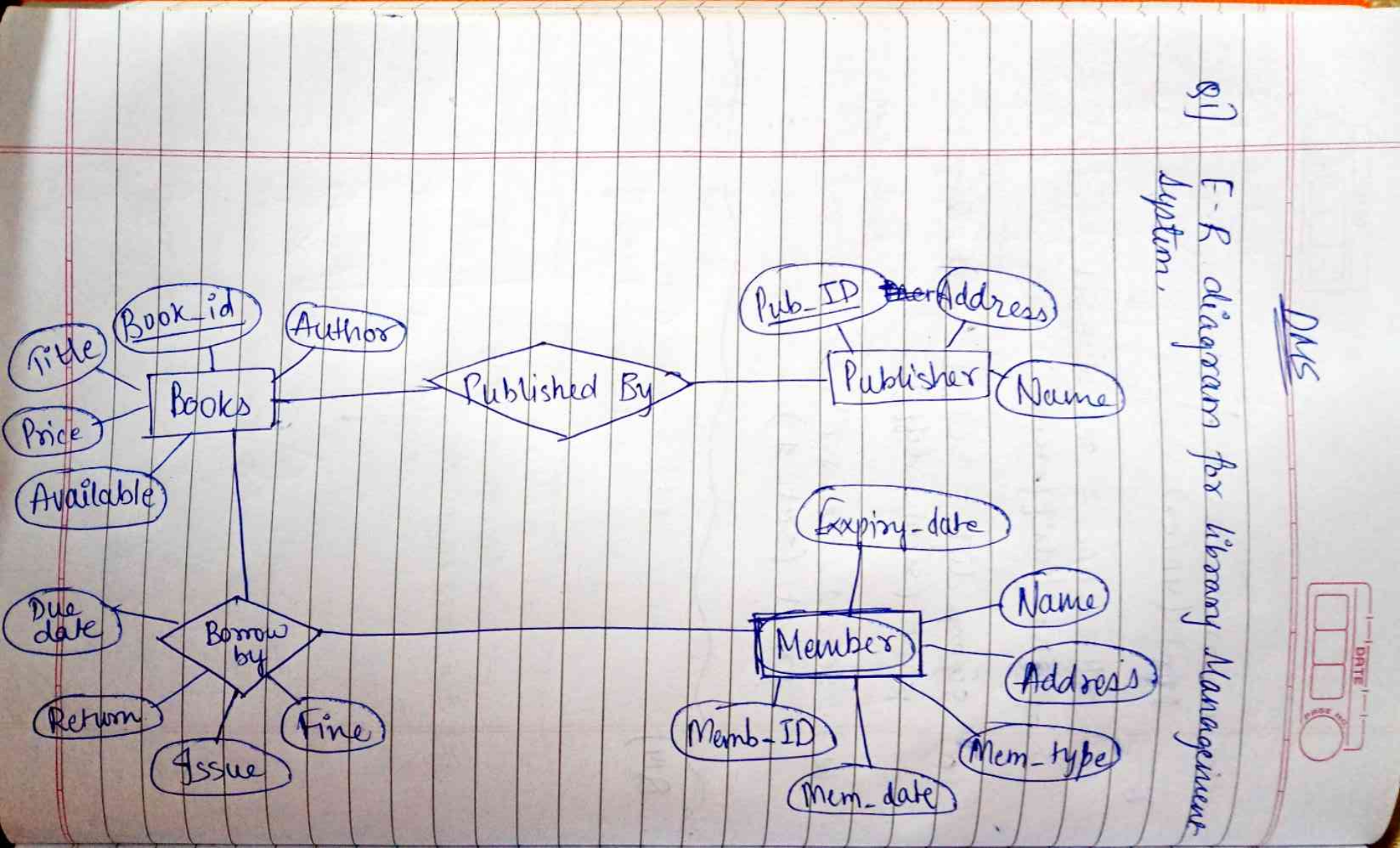
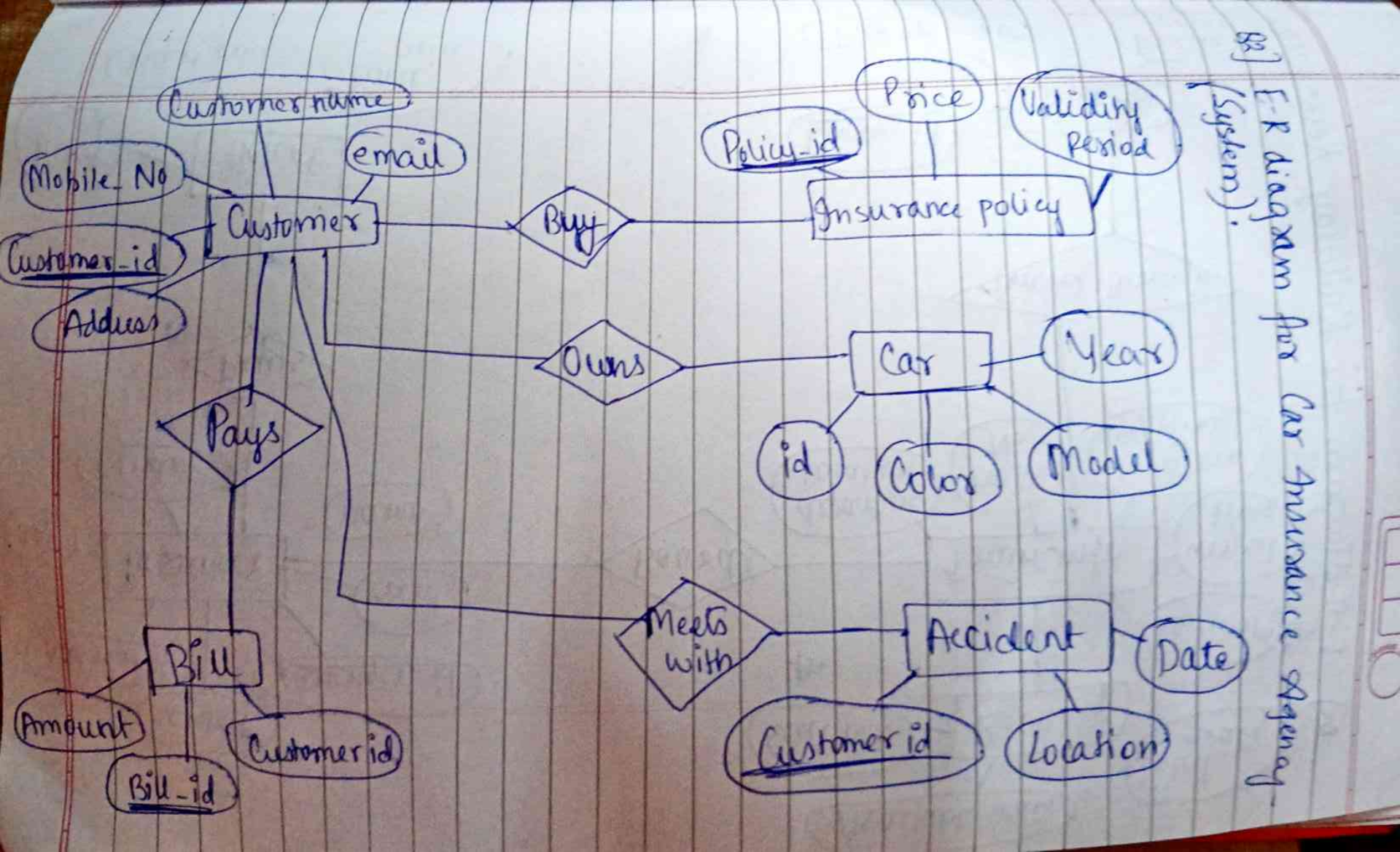
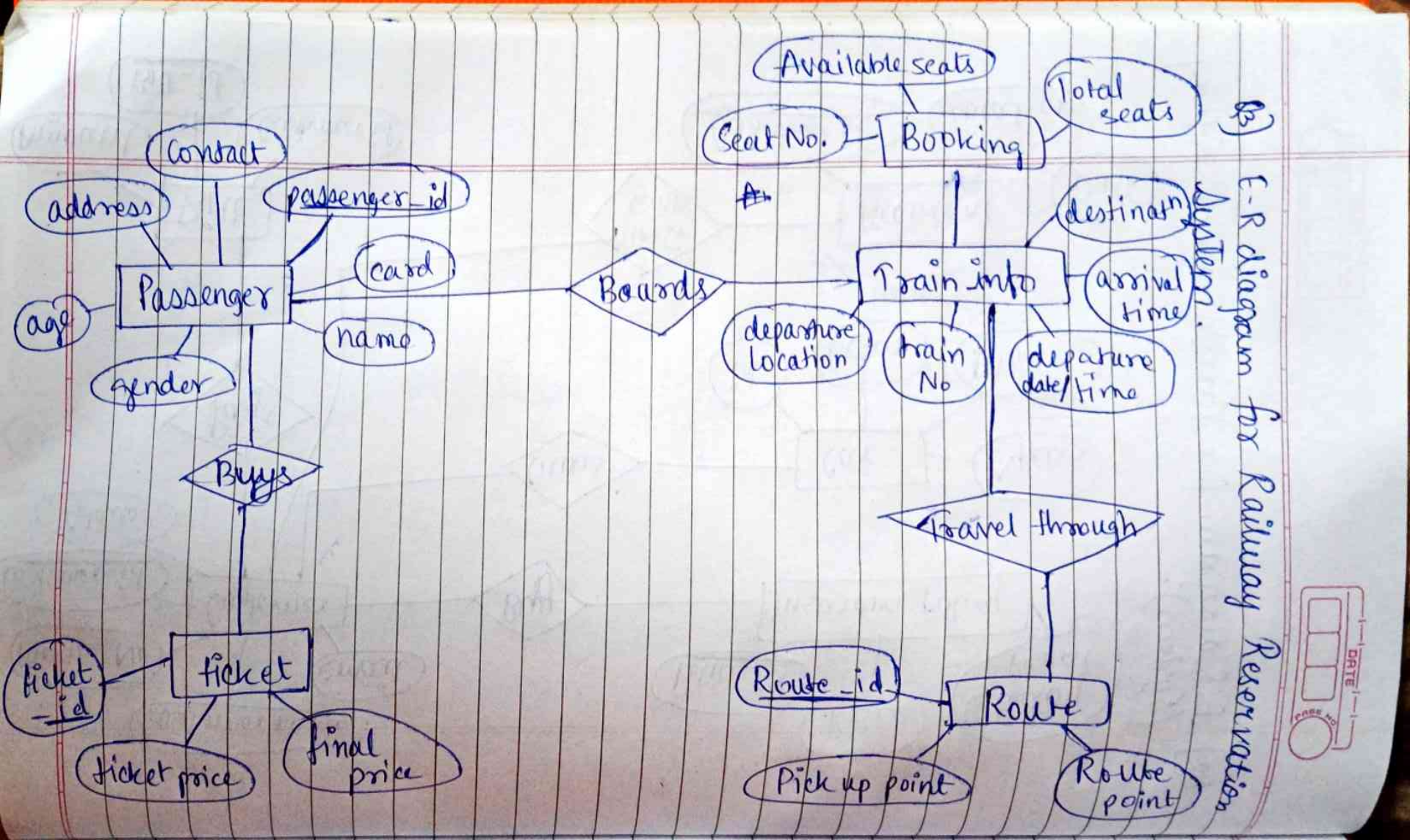
DMS

Q1) E-R diagram for library Management system.



Q2] E-R diagram for Car Insurance Agency System.




E-R diagram for Railway Reservation System.
Passenger
address
Contact
passenger_id
card
name
gender
age
Buys
ticket
ticket_id
ticket price
final price
Boards
Train info
Available seats
Total seats
Booking
Seat No.
destination
arrival time
departure location
train No
departure date/time
Travel through
Route
Route_id
Pick up point
Route point

Q4] Create and execute DDL commands using SQL.

⇒ DDL commands are :-
Create, Alter, Drop, Truncate.

- Create :-

⇒ Create table employee (Id integer, emp_name varchar(20), hire_date date);

- Alter :-

⇒ Alter table employee
ADD city varchar(20);

- Drop :-

⇒ Drop table employee;

- Truncate :-

⇒ Truncate table employees;

---

Q5] Apply following contraints on table employee,

(i) Primary Key (ii) Not Null
(iii) Check constraint.

(i) Primary Key :-

⇒ Create table employee (Id integer primary key, emp-name varchar (20), hire_date date);

(ii) Not Null :-

⇒ Create table employee (Id integer, emp_name varchar (20) NOT NULL hire_date date);

(iii) Check constraint :-

⇒ Create table employee (Id integer, emp-name, emp-age INT check (emp_age > 18));

---

Q6) Create & execute DML commands using SQL.

⇒ DML commands are :-
Insert, Update, Delete, Select.

(i) Insert :-

⇒ Create table employee
(emp-id INT primary key, emp-name
varchar(100) NOT NULL, emp-age
check (emp-age > 18));

Insert into employee (emp-id, emp-name,
emp-age) values (1, 'John', 30),
(2, 'Smith', 25);

(ii) Update :-

⇒ Update employee set emp-name =
'Johnathan Doe', emp-age = 32
where emp-id = 1;

(iii) Select :-

⇒ Select * from employee;

(iv) Delete :-

⇒ Delete from employee
where emp-id = 2;

Q7] Write queries using following operators

(i) Range searching operator

⇒ Between :-
Select * from employee where emp_age between 20 AND 30;

(ii) Pattern Matching Operator.

⇒ Like :-
Select * from employee where emp_name like 'J%';

---

Q8] Write queries using following functions
(i) String (ii) Arithmetic.

⇒ (i) String :-

- Upper :- Select emp_id UPPER (emp_name) AS upper_name FROM employee;

- Substring or LEFT :-

Select emp_id, SUBSTRING (emp_name, 1, 3) AS short_name from employee ;

OR

Select emp_id, LEFT (emp_name, 3) AS short_name from employee ;

- LENGTH :- Select emp_id LENGTH (emp_name) AS name_length from employee ;

- CONCAT :- Select emp_id CONCAT emp_name, '- Employee') AS full_description from employee;

(ii) Arithmetic :-

- Add :- Select emp_id, emp_name, emp_age + 5 AS age_in_5_years from employee ;

- Power :- Select emp_id, emp_name, POWER (emp_age, 2) AS age_squared from employee ;

- AVG :- Select AVG (emp_age) AS average_age from employee;
- MOD :- Select emp_id, emp_name, MOD (emp_age, 10) AS remainder from employee;

---

Q9] Write queries using following functions
(i) Date & Time
(ii) Aggregate funct^n

⇒

(i) Date & Time :-

- NOW() :- Selet NOW() AS current_datetime;
- YEAR() :- Select emp_id, emp_name, YEAR (hire_date) AS hire_year from employee;
- DATE_ADD() :- Select emp_id, emp_name, DATE_ADD (hire_date, INTERVAL 30 DAY) AS probation_end_date from employee;

(ii) Aggregate function :-

- COUNT :- Select count(*) AS total_employees from employee;
- MAX :- Select MAX (emp_age) AS max_age from employee;
- MIN :- Select MIN (emp_age) AS min_age from employee;
- SUM :- Select SUM (emp_age) AS total_age from employee;

---

Q10] Execute queries using select command with (i) WHERE clause.
(ii) HAVING clause.

→ (i) WHERE clause :- Select emp_id, emp_name, salary from employee WHERE salary > 50000;

(ii) HAVING clause :- Select dept_id, AVG (salary) AS avg_salary from employee group by dept_id HAVING AVG (salary) > 60000;