

G. H. RAISONI COLLEGE OF ENGINEERING, NAGPUR
(An Autonomous Institute affiliated to RTM Nagpur University)

Department of Computer Science & Engineering

Date of Submission: 22/08/20

Name of Subject: Database Management Systems
Session: 2020-21

Student Details:

Roll Number	4 & 5
Member Name 1	Koppiseti Shreyasi
Member Name 2	Kshitija Chilbule
Semester	III
Section	B
Branch	CSE

**MINI Project
Report**

Name of Project: Hotel management system

Introduction:

The software product Hotel management system (HMS) is developed to automate major operations of the hotel. This system will help in managing the data of different departments of the hotel, thus it will simplify the day to day tasks of the end users and they will be able to handle customer services with efficiency.

The HMS consist of total five modules, each module is briefly described below:

1. Room booking module:

This module contains data of rooms, customers and details of payment made by customers. The Sub modules are as follows:

- a) Room information
- b) Room tariff
- c) Customer information
- d) Customer payment record

Room data consist of room number, type, tariff and room status (whether room is empty, occupied or under maintenance). Customer data consist of customer id, name, address, check – in date and time, check – out date and time, duration of stay, valid ID,

mobile number, booked room type and room number. Customer payment record consist of receipt number, customer id, payment date, payment mode and amount paid by the customer.

2. Employee personnel module:

This module contains data of employees of the hotel and leaves taken by employees. The sub modules are as follows:

- a) Employee information
- b) Employee leaves

Employee data consist of employee id, name, date of birth, date of joining, basic salary and bank details. Leaves record consist of employee id, leave type, duration of leave, joining date and date of last leave taken.

3. Store module:

This module contains data of stock items and capital inventory. The sub modules are as follows:

- a) Stock item details
- b) Capital inventory

Stock items data consists of item details like item code, name, rate, quantity, last date of purchase and minimum stock quantity of item. Capital inventory data consists of inventory code, inventory name, brand, and maintenance duration and installation date.

4. Maintenance module:

This module contains a record of maintenance of inventory items. Maintenance data consist of inventory code of item, break down date, restoration date and repair charges.

5. Purchase module:

This module contains data of stock items purchased by the hotel. Purchase data consist order number, item code, order date, order quantity, order amount and delivery date.

ER Diagram:

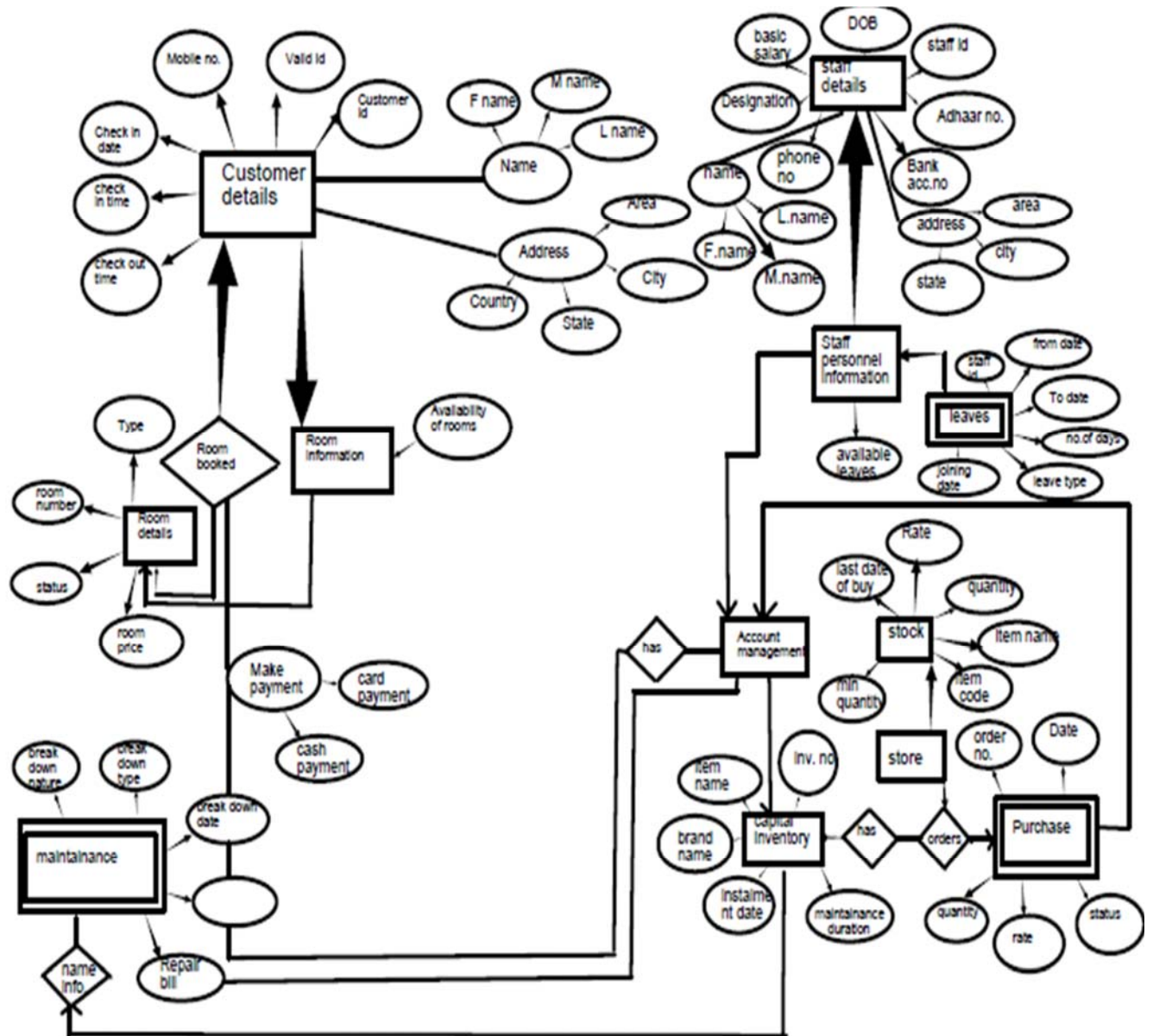


Table Syntax:

Following are the syntax of SQL commands used in Hotel management system.

1. Create Table:

This command is used to create tables of database. The syntax is as follows,

```
CREATE TABLE table_name
(
    Column_1_name data type (size),
    Column_2_name data type (size),
    Column_3_name data type (size),
    Column_4_name data type (size),
);
```

2. Primary Key:

Primary key is an attribute/column of the table that can identify each record in the table uniquely.

```
PRIMARY KEY (column_name);
```

3. Foreign Key:

Foreign is a column or group of columns of the table that references the primary key of another table. It helps in maintaining referential integrity between the child and parent table. The syntax is as follows,

```
FOREIGN KEY (column_name)
REFERENCES table_name (name of the column which is primary key of reference table);
```

We can also specify the action on delete and on update in foreign key.

The actions are:

- | | | |
|---------------|-----------------|---------------|
| i. SET NULL | ii. SET DEFAULT | iii. RESTRICT |
| iv. NO ACTION | v. CASCADE | |

4. Check constraint:

This constraint is used to limit the values of attribute / column. It allows only those values to be recorded in column, which satisfy the condition specified in CHECK (). The syntax is as follows:

```
column_name data type (size) CHECK (condition),
```

5. Not null constraint:

This constraint checks that the value for that attribute should not be null. The syntax is as follows:

```
CREATE TABLE table_name
(
    Column_1_name data type (size) CHECK (condition),
    Column_2_name data type (size),
    Column_3_name data type (size) CHECK (condition),
```

Column_4_name data type (size),);

6. Unique constraint:

This constraint ensures that all the values in that column are unique. This column can also be a primary key. The syntax is as follows:

```
CREATE TABLE table_name
(
Column_1_name data type (size),
Column_2_name data type (size),
Column_3_name data type (size),
Column_4_name data type (size) UNIQUE,
);
```

7. Alter:

Alter command is used to alter the structure of table in data base. It can be used to add, modify and delete column. The syntax is as follows,

i) ADD

```
ALTER TABLE table_name ADD column_name data type (size);
```

ii) MODIFY or ALTER COLUMN

```
ALTER TABLE table_name MODIFY column_name TYPE data type (size);
```

Or

```
ALTER TABLE table_name ALTER COLUMN column_name TYPE data type
(size);
```

iii) DROP

```
ALTER TABLE table_name DROP column_name;
```

8. Drop:

Drop command is used to drop a structure. Structure can be a column of a table or whole table or a database. The syntax is as follows,

```
DROP TABLE structure_name;
```

9. Insert:

Insert command is used to enter data into the tables of the database. The syntax is as follows,

```
INSERT INTO table_name VALUES (value_1, value_2, value_3 . . .);
```

Or

```
INSERT INTO table_name (column_1, column_2, column_3...) VALUES
(value_1, value_2, value_3 . . .);
```

9. Update:

Update command is used to modify data in a table. The syntax is as follows,

UPDATE table_name SET column_name = value WHERE condition;

10. Delete:

Delete command is used to delete one or more rows in a table. The syntax is as follows,

DELTE FROM table_name WHERE condition;

11. Truncate table:

Truncate table statement is used to delete all the data form a table. The syntax is as follows,

TRUNCATE TABLE table_name;

12. Select:

Select statement is used to query or retrieve data form a table. The syntax to retrieve all the columns of the table is as follows,

SELECT * FROM table_name;

Select statement has many clauses, some them which are used in this hotel management system are as follows:

- | | |
|---------------|-----------|
| i. DISTINCT | ii. WHERE |
| iii. GROUP BY | iv. JOINS |

13. Distinct clause:

The distinct clause is used in select statement to remove duplicate rows from result set. The syntax is as follows,

SELECT DISTINCT column_1, column_2 ... FROM table_name;

14. Where clause:

Where clause is used to filter rows. The syntax is as follows:

SELECT column_name FROM table_name WHERE (condition);

15. IN operator:

IN operator in where clause is used to check if a value matches with value in the list of values.

VALUE IN (value_1, value_2);

or

VALUE IN (SELECT column_name FROM table_name);

16. BETWEEN operator:

BETWEEN operator is used to match a value in a range of values. The syntax is as

follows,

Value BETWEEN low AND high;

17. LIKE operator:

LIKE operator is used to match string patterns with strings in a table. Patterns are constructed by combining literal values with wildcard characters. The wildcard characters are as follows:

- i. percentage sign (%) it matches any sequence of zero or more characters.
- ii. Underscore sign (_) matches any single character.

The syntax is as follows:

Value LIKE pattern;

Value LIKE 'a%b';

Value LIKE '__a';

18. Logical operators:

Logical operators return the values for which the condition is true. Following are the logical operators in SQL:

1. OR – Returns value if either condition is true.
2. AND – Returns value if all the conditions are true.
3. NOT –Return the opposite value.

The syntax is as follows:

SELECT column_name FROM table_name WHERE column_name LOGICAL
OPERATOR value;

19. Aggregate functions:

Aggregate functions perform a calculation on a set of rows and return a single row.

Aggregate functions in SQL are as follows:

1. MAX () – Return the maximum value.
2. MIN () – Return the minimum value.
3. COUNT () – Return the number of values.
4. AVG () – Return the average value.
5. SUM () –Return the sum of all or distinct values.

The syntax is as follows:

SELECT AGGREGATE FUNCTION (column_name) FROM table_name;

20. GROUP BY clause:

The GROUP BY clause divides the rows returned from the SELECT statement into groups. For each group, you can apply an aggregate function e.g., SUM () to calculate the sum of items or COUNT () to get the number of items in the groups. Syntax is as follows:

SELECT column_name FROM table_name GROUP BY column_name;

21. JOINS:

Join is used to combine columns from two or more tables based on the values of the common columns between related tables. The common columns are typically the primary key columns of the first table and foreign key columns of the second table.

Types of JOINS

1. INNER JOIN: In this type of join, the values of the common columns are compared. If the values are equal, then inner join creates a new row that contains columns from both tables and adds this new row to the result set. Following is the syntax.

```
SELECT (column_1, column_2 . . .)
FROM table_1 INNER JOIN table_2 ON col_table_1 = col_table_2;
```

2. LEFT JOIN: In this type of join, values of the common column are compared. If the values are equal then left join creates a new row that contains columns from both tables and adds this new row to result set. If the values are not equal, then left join creates a new row that contains columns from both tables and adds this new row to result set. However, it fills the columns of the right table with null or is left blank. The syntax is as follows.

```
SELECT (column_1, column_2 . . .)
FROM table_1 LEFT JOIN table_2 ON col_table_1 = col_table_2;
```

3. RIGHT JOIN: This is opposite of left join. It starts selecting data from right table and compares values. If the values are equal then right join creates a new row that contains columns from both tables and adds this new row to result set. If the values are not equal, then right join creates a new row that contains columns from both tables and adds this new row to result set. However, it fills the columns of the left table with null or is left blank. The syntax is as follows.

```
SELECT (column_1, column_2 . . .)
FROM table_1 RIGHT JOIN table_2 ON col_table_1 = col_table_2;
```

4. FULL JOIN: full join returns a result set that contains all rows from both left and right tables, with the matching rows from both sides if available. In case there is no match, the columns of the table will be filled with NULL.

```
SELECT (column_1, column_2 . . .)
FROM table_1 FULL JOIN table_2 ON col_table_1 = col_table_2;
```

5. NATURAL JOIN: A natural join is a join that creates an implicit join based on the same column names in the joined tables. The syntax is as follows.

```
SELECT select_list FROM table_1 NATURAL JOIN table_2;
```


22. Column aliases:

Column alias allows you to assign a table or column or an expression in select list of select statement a temporary name. The syntax is as follows:

```
SELECT column_name AS alias_name FROM table_name;
```

Or

```
SELECT alias_name_1.column_1, alias_name_2.column_name_2 FROM table_1  
AS alias_name_1, table_2 AS alias_name_2;
```

23. Sub query:

A subquery is a query nested inside another query such as SELECT, INSERT, DELETE and UPDATE. Subquery is used to return data that will be used in outer query as a condition to further restrict the data to be retrieved. Syntax is as follows:

```
SELECT (column_1, column_2 . . .) FROM table_1 where column_name =  
(SELECT column_name FROM table_2);
```

24. ANY operator:

ANY operator compares a value to a values to a set of values returned by subquery. The syntax is as follows:

```
Expression ANY (subquery);
```

```
SELECT column_name FROM table_name WHERE column_name = ANY (sub  
query);
```

25. Commit statement:

Commit command is used to permanently save any transaction into the database. When we use any DML command like insert, update or delete, the changes made by these commands are not permanent, until the current session is closed, the changes made by these commands can be rolled back.

To avoid that, we use the commit command to mark the changes as permanent. The syntax is as follows:

```
commit;
```

26. Savepoint statement:

savepoint command is used to temporarily save a transaction so that you can roll back to that point whenever required. Following is savepoint command's syntax,

```
savepoint savepoint_name;
```

27. Rollback statement:

This command restores the database to last committed state. It is also used with `savepoint` command to jump to a savepoint in an ongoing transaction. If we have used the `update` command to make some changes into the database, and realise that those changes were not required, then we can use the `rollback` command to rollback those changes, if they were not committed using the `commit` command. The syntax is as follows:

```
rollback;
```

Or

```
rollback to savepoint name;
```

SQL Script:

Module 1: Room booking system

a) Room information table

```
create table room_info(
    room_no    int NOT NULL,
    room_type  char(20),
    room_status char(2) CHECK (room_status in('E','O','UM')),
    PRIMARY KEY(room_no)
);

insert into room_info values(101,'NASB','E');
insert into room_info values(102,'NASB','E');
insert into room_info values(103,'NASB','E');
insert into room_info values(104,'NASB','E');
insert into room_info values(105,'NASB','E');
insert into room_info values(106,'NASB','E');
insert into room_info values(201,'NADB','E');
insert into room_info values(202,'NADB','E');
insert into room_info values(203,'NADB','E');
insert into room_info values(204,'NADB','E');
insert into room_info values(205,'NADB','E');
insert into room_info values(206,'NADB','E');
insert into room_info values(301,'ACSB','E');
insert into room_info values(302,'ACSB','E');
insert into room_info values(303,'ACSB','E');
insert into room_info values(304,'ACSB','E');
insert into room_info values(305,'ACSB','E');
insert into room_info values(306,'ACSB','E');
insert into room_info values(401,'ACDB','E');
insert into room_info values(402,'ACDB','E');
insert into room_info values(403,'ACDB','E');
insert into room_info values(404,'ACDB','E');
```

```

insert into room_info values(405,'ACDB','E');
insert into room_info values(406,'ACDB','E');
insert into room_info values(501,'JS','E');
insert into room_info values(502,'JS','E');
insert into room_info values(503,'JS','E');
insert into room_info values(504,'MS','E');
insert into room_info values(505,'MS','E');
insert into room_info values(506,'MS','E');

```

```

update room_info set room_status = 'O' where room_no
in(101,102,106,203,205,301,405,502);
update room_info set room_status = 'O' where room_no = 201;
select * from room_info;

```

Output pane

Data Output	Explain	Messages	History
room_no integer	room_type character(20)	room_status character(2)	
1	103 NASB	E	
2	104 NASB	E	
3	105 NASB	E	
4	202 NADB	E	
5	204 NADB	E	
6	206 NADB	E	
7	302 ACSB	E	
8	303 ACSB	E	
9	304 ACSB	E	
10	305 ACSB	E	
11	306 ACSB	E	
12	401 ACDB	E	
13	402 ACDB	E	
14	403 ACDB	E	
15	404 ACDB	E	
16	406 ACDB	E	
17	501 JS	E	
18	503 JS	E	
19	504 MS	E	
20	505 MS	E	
21	506 MS	E	
22	101 NASB	O	
23	102 NASB	O	
24	106 NASB	O	
25	203 NADB	O	
26	205 NADB	O	
27	301 ACSB	O	
28	405 ACDB	O	
29	502 JS	O	
30	201 NADB	O	

```

select count(room_type) from room_info where room_type = 'NASB';
select count(room_type) from room_info where room_type = 'NADB';
select count(room_type) from room_info where room_type = 'ACSB';
select count(room_type) from room_info where room_type = 'ACDB';

```

Output pane

Data Output	Explain	Messages	History
count bigint			
1	6		

b) Room tariff table

```
create table room_t
(
  room_type    char(4) primary key,
  room_des     char(20),
  room_tariff  numeric(4,0)CHECK(room_tariff BETWEEN 3000 AND 9000)
);
insert into room_t values('ACDB','AC Double bed',4500);
insert into room_t values('NADB','NON AC Double bed',3500);
insert into room_t values('ACSB','AC Single bed',4000);
insert into room_t values('NASB','NON AC Single bed',3000);
insert into room_t values('MS','Master suite',8000);
insert into room_t values('JS','Junior suite',6000);
select * from room_t
```

Output pane			
Data Output			
	room_type character(4)	room_des character(20)	room_tariff numeric(4,0)
1	NASB	NON AC Single bed	3000
2	ACSB	AC Single bed	3500
3	NADB	NON AC Double bed	4000
4	ACDB	AC Double bed	4500
5	MS	Master suite	8000
6	JS	Junior suite	6000

```
select max(room_tariff) from room_t;
```

Output pane		
Data Output		
	max numeric	
1	8000	

```
select min(room_tariff) from room_t;
```

Output pane		
Data Output		
	min numeric	
1	3000	

```
select count(room_type) from room_info where room_type = 'JS';
select count(room_type) from room_info where room_type = 'MS';
```

Output pane		
Data Output		
	count bigint	
1	3	

```
select ri.room_no, ri.room_type, rt.room_des, rt.room_tariff from room_info ri, room_t
rt where ri.room_type = rt.room_type;
```

Output pane

Data Output	Explain	Messages	History	
	room_no integer	room_type character(20)	room_des character(20)	room_tariff numeric(4,0)
1	103	NASB	NON AC Single bed	3000
2	104	NASB	NON AC Single bed	3000
3	105	NASB	NON AC Single bed	3000
4	202	NADB	NON AC Double bed	4000
5	204	NADB	NON AC Double bed	4000
6	206	NADB	NON AC Double bed	4000
7	302	ACSB	AC Single bed	3500
8	303	ACSB	AC Single bed	3500
9	304	ACSB	AC Single bed	3500
10	305	ACSB	AC Single bed	3500
11	306	ACSB	AC Single bed	3500
12	401	ACDB	AC Double bed	4500
13	402	ACDB	AC Double bed	4500
14	403	ACDB	AC Double bed	4500
15	404	ACDB	AC Double bed	4500
16	406	ACDB	AC Double bed	4500
17	501	JS	Junior suite	6000
18	503	JS	Junior suite	6000
19	504	MS	Master suite	8000
20	505	MS	Master suite	8000
21	506	MS	Master suite	8000
22	101	NASB	NON AC Single bed	3000
23	102	NASB	NON AC Single bed	3000
24	106	NASB	NON AC Single bed	3000
25	203	NADB	NON AC Double bed	4000
26	205	NADB	NON AC Double bed	4000
27	301	ACSB	AC Single bed	3500
28	405	ACDB	AC Double bed	4500
29	502	JS	Junior suite	6000
30	201	NADB	NON AC Double bed	4000

c) Customer information table

```
create table customer_info(
    cust_id    int NOT NULL,
    cust_name  char(40) NOT NULL,
    mobile_no  int NOT NULL,
    address_1  varchar(30),
    address_2  varchar(30),
    address_3  varchar(30),
    check_in_dt date NOT NULL,
    check_in_tm time NOT NULL,
    check_out_dt date NOT NULL,
    check_out_tm time NOT NULL ,
    No_of_days int,
    ID         varchar(10),
    PRIMARY KEY(cust_id)
);
```

```
alter table customer_info alter column check_out_dt set default NULL;
alter table customer_info alter column check_out_tm set default NULL;
alter table customer_info drop column mobile_no;
alter table customer_info add column mobile_no bigint;
alter table customer_info add column room_ty char(4);
alter table customer_info add column room_number int;
update customer_info set room_number = 401 where cust_id = 10126;
```

```

update customer_info set room_number = 403 where cust_id = 10117;
insert into customer_info
(cust_id,cust_name,mobile_no,address_1,address_2,address_3,room_ty,check_in_dt,che
ck_in_tm,check_out_dt,check_out_tm,no_of_days,ID,room_number)values(10112,'Vaib
havi Apte','34562310','Prakash Enclave Prakash nagar ','Nagpur Maharastra','India
','NASB','2020-3-20','11:10:10','2020-3-24','11:10:10', 2 ,'Aadhar',103);
insert into customer_info
(cust_id,cust_name,mobile_no,address_1,address_2,address_3,room_ty,check_in_dt,che
ck_in_tm,check_out_dt,check_out_tm,no_of_days,ID,room_number)values(10113,'Vaib
hav Shukla','1047834777','Om Enclave Ulas nagar ','Kanpur UP','India ','NASB','2020-
3-25','10:10:10','2020-3-30','12:10:10', 5 ,'Aadhar',101);
insert into customer_info
(cust_id,cust_name,mobile_no,address_1,address_2,address_3,room_ty,check_in_dt,che
ck_in_tm,check_out_dt,check_out_tm,no_of_days,ID,room_number)values(10114,'Kris
hna Pandy','2035673490','Prakash Enclave Prakash nagar ','Nagpur Maharastra','India
','ACSB','2020-4-1','9:15:10','2020-4-3','11:10:10', 3 ,'Passport',302);
insert into customer_info
(cust_id,cust_name,mobile_no,address_1,address_2,address_3,room_ty,check_in_dt,che
ck_in_tm,check_out_dt,check_out_tm,no_of_days,ID,room_number)values(10115,'Sanj
ay Singh','2035672491','plt no 201 Mani nagar ','Varodara Gujrat','India ','NADB','2020-
4-2','9:15:10','2020-4-4','9:10:10', 2 ,'Ration card',202);
insert into customer_info
(cust_id,cust_name,mobile_no,address_1,address_2,address_3,room_ty,check_in_dt,che
ck_in_tm,check_out_dt,check_out_tm,no_of_days,ID,room_number)values(10116,'Sanj
ana Pandy','3033673590','plt no 4 Laxmi nagar ','Goa Maharastra','India ','NADB','2020-
4-5','7:15:10','2020-4-7','11:10:10', 2 ,'Passport',203);
insert into customer_info
(cust_id,cust_name,mobile_no,address_1,address_2,address_3,room_ty,check_in_dt,che
ck_in_tm,check_out_dt,check_out_tm,no_of_days,ID,room_number)values(10117,'Kish
ore Kumar','8835673490','Mannat Gandhi nagar ','Rajkot Gujarat','India ','ACDB','2020-
4-9','6:20:10','2020-4-13','11:10:10', 5 ,'Aadhar card',401);
insert into customer_info
(cust_id,cust_name,mobile_no,address_1,address_2,address_3,room_ty,check_in_dt,che
ck_in_tm,check_out_dt,check_out_tm,no_of_days,ID,room_number)values(10118,'Ash
ok Malviya','8895673490','Manish colony Ganesh nagar ','Bhopal Madhya
Pradesh','India ','ACDB','2020-4-9','6:25:10','2020-4-13','11:10:10', 5 ,'Aadhar
card',402);
insert into customer_info
(cust_id,cust_name,mobile_no,address_1,address_2,address_3,room_ty,check_in_dt,che
ck_in_tm,check_out_dt,check_out_tm,no_of_days,ID,room_number)values(10119,'Kish
ore Agarwal','9955673490','defence colony kokila nagar ','Goa Goa','India ','JS','2020-4-
9','6:30:10','2020-4-13','11:10:10', 5 ,'Aadhar card',501);
insert into customer_info
(cust_id,cust_name,mobile_no,address_1,address_2,address_3,room_ty,check_in_dt,che
ck_in_tm,check_out_dt,check_out_tm,no_of_days,ID,room_number)values(10120,'Man
ik Kumar','8535673490','Pratap Enclave Pratap nagar ','Hyderabad Telangana','India
','ACDB','2020-4-9','6:20:10','2020-4-13','11:10:10', 5 ,'Aadhar card',405);

```

```

insert into customer_info
(cust_id,cust_name,mobile_no,address_1,address_2,address_3,room_ty,check_in_dt,che
ck_in_tm,check_out_dt,check_out_tm,no_of_days,ID,room_number)values(10121,'Prat
ap Bansal','8835573490','Manaya roofs Alok nagar ','Nagpur Maharastra','India
','ACSB','2020-4-8','6:20:10','2020-4-11','11:10:10', 4 ,'Aadhar card',303);
insert into customer_info
(cust_id,cust_name,mobile_no,address_1,address_2,address_3,room_ty,check_in_dt,che
ck_in_tm,check_out_dt,check_out_tm,no_of_days,ID,room_number)values(10122,'Anu
j Lulu','99562310','Prakash Enclave Prakash nagar ','Nagpur Maharastra','India ','ACSB',
'2020-3-20','11:10:10','2020-3-24','11:10:10', 2 ,'Aadhar',301);
insert into customer_info
(cust_id,cust_name,mobile_no,address_1,address_2,address_3,room_ty,check_in_dt,che
ck_in_tm,check_out_dt,check_out_tm,no_of_days,ID,room_number)values(10123,'Sidh
ardh Shukla','9947834777','Om Enclave Ulas nagar ','Kanpur UP','India ','NADB', '2020-
3-25','10:10:10','2020-3-30','12:10:10', 5 ,'Aadhar',205);
insert into customer_info
(cust_id,cust_name,mobile_no,address_1,address_2,address_3,room_ty,check_in_dt,che
ck_in_tm,check_out_dt,check_out_tm,no_of_days,ID,room_number)values(10124,'Vija
y Kumar Pandey','6735673490','Prakash Enclave Prakash nagar ','Nagpur
Maharastra','India ','JS','2020-4-1','9:15:10','2020-4-3','11:10:10', 3 ,'Passport',502);
insert into customer_info
(cust_id,cust_name,mobile_no,address_1,address_2,address_3,room_ty,check_in_dt,che
ck_in_tm,check_out_dt,check_out_tm,no_of_days,ID,room_number)values(10125,'Kart
ar Singh','9835672491','plt no 201 Mani nagar ','Varodara Gujrat','India ','NASB', '2020-
4-2','9:15:10','2020-4-4','9:10:10', 2 ,'Ration card',102);
insert into customer_info
(cust_id,cust_name,mobile_no,address_1,address_2,address_3,room_ty,check_in_dt,che
ck_in_tm,check_out_dt,check_out_tm,no_of_days,ID,room_number)values(10126,'Pula
kit Samrat','7733673590','plt no 4 Laxmi nagar ','Goa Maharastra','India ','ACDB', '2020-
4-5','7:15:10','2020-4-7','11:10:10', 2 ,'Passport',104);
insert into customer_info
(cust_id,cust_name,mobile_no,address_1,address_2,address_3,room_ty,check_in_dt,che
ck_in_tm,check_out_dt,check_out_tm,no_of_days,ID,room_number)values(10127,'Dee
pen Chaturvedi','9935673490','Mannat Gandhi nagar ','Rajkot Gujarat','India ','NASB',
'2020-4-9','6:20:10','2020-4-13','11:10:10', 5 ,'Aadhar card',105);
insert into customer_info
(cust_id,cust_name,mobile_no,address_1,address_2,address_3,room_ty,check_in_dt,che
ck_in_tm,check_out_dt,check_out_tm,no_of_days,ID,room_number)values(10128,'Mall
ika Malviya','5895673490','Manish colony Ganesh nagar ','Bhopal Madhya
Pradesh','India ','NADB', '2020-4-9','6:25:10','2020-4-13','11:10:10', 5 ,'Aadhar
card',201);
insert into customer_info
(cust_id,cust_name,mobile_no,address_1,address_2,address_3,room_ty,check_in_dt,che
ck_in_tm,check_out_dt,check_out_tm,no_of_days,ID,room_number)values(10129,'Push
pa Agarwal','9755673490','defence colony kokila nagar ','Goa Goa','India ','NASB',
'2020-4-9','6:30:10','2020-4-13','11:10:10', 5 ,'Aadhar card',106);
insert into customer_info

```



```
(cust_id,cust_name,mobile_no,address_1,address_2,address_3,room_ty,check_in_dt,check_in_tm,check_out_dt,check_out_tm,no_of_days,ID,room_number)values(10130,'Manisha Kumari','8935673490','Pratap Enclave Pratap nagar ','Hyderabad Telangana','India','NADB','2020-4-9', '6:20:10','2020-4-13','11:10:10', 5 ,'Aadhar card',204);
```

```
insert into customer_info
```

```
(cust_id,cust_name,mobile_no,address_1,address_2,address_3,room_ty,check_in_dt,check_in_tm,check_out_dt,check_out_tm,no_of_days,ID,room_number)values(10111,'Kani ka Kapoor','9835573490','Manaya roofs Alok nagar ','Nagpur Maharastra','India','MS','2020-4-8', '6:20:10','2020-4-11','11:10:10', 4 ,'Aadhar card',505);
```

```
select * from customer_info;
```

Output pane													
Data Output Explain Messages History													
	cust_id	cust_name	address_1	address_2	address_3	check_in_dt	check_in_tm	check_out_dt	check_out_tm	no_of_days	id	mobile_no	room_number
	integer	character(40)	character varying(30)	character varying(30)	character varying(30)	date	time without time zone	date	time without time zone	integer	character(15)	bigint	integer
1	10114	Krishna Pand	Prakash Enclave Pr	Nagpur Maharastra	India	2020-04-01	09:15:10	2020-04-03	11:10:10	3	Passport	935673490	302
2	10115	Sanjay Sing	plc no 201 Mani na	Varodara Gujrat	India	2020-04-02	09:15:10	2020-04-04	09:10:10	2	Ration card	935672491	202
3	10116	Sanjana Pand	plc no 4 Laxmi nag	Goa Maharastra	India	2020-04-05	07:15:10	2020-04-07	11:10:10	2	Passport	933673590	203
4	10118	Ashok Malvi	Manish colony Gene	Rhopal Madhya Prad	India	2020-04-09	06:25:10	2020-04-13	11:10:10	5	Aadhar card	955673490	402
5	10119	Kishore Agar	defence colony kok	Goa Goa	India	2020-04-09	06:30:10	2020-04-13	11:10:10	5	Aadhar card	955673490	501
6	10120	Manik Kumar	Pratap Enclave Pra	Hyderabad Telangan	India	2020-04-09	06:20:10	2020-04-13	11:10:10	5	Aadhar card	935673490	405
7	10121	Pratap Bans	Manaya roofs Alok	Nagpur Maharastra	India	2020-04-08	06:20:10	2020-04-11	11:10:10	4	Aadhar card	935573490	303
8	10122	Anuj Lulu	Prakash Enclave Pr	Nagpur Maharastra	India	2020-03-20	11:10:10	2020-03-24	11:10:10	2	Aadhar	99562310	301
9	10123	Sidhardh Shi	Om Enclave Ulas na	Kanpur UP	India	2020-03-25	10:10:10	2020-03-30	12:10:10	5	Aadhar	947834777	205
10	10124	Vijay Kumar	Prakash Enclave Pr	Nagpur Maharastra	India	2020-04-01	09:15:10	2020-04-03	11:10:10	3	Passport	735673490	502
11	10125	Karter Sing	plc no 201 Mani na	Varodara Gujrat	India	2020-04-02	09:15:10	2020-04-04	09:10:10	2	Ration card	935672491	102
12	10127	Deepen Chat	Mannat Gandhi naga	Rajkot Gujarat	India	2020-04-09	06:20:10	2020-04-13	11:10:10	5	Aadhar card	935673490	105
13	10128	Mallika Mal	Manish colony Gene	Rhopal Madhya Prad	India	2020-04-09	06:25:10	2020-04-13	11:10:10	5	Aadhar card	955673490	201
14	10129	Pushpa Agar	defence colony kok	Goa Goa	India	2020-04-09	06:30:10	2020-04-13	11:10:10	5	Aadhar card	755673490	106
15	10130	Manisha Kum	Pratap Enclave Pra	Hyderabad Telangan	India	2020-04-09	06:20:10	2020-04-13	11:10:10	5	Aadhar card	935673490	204
16	10111	Kanika kapo	Manaya roofs Alok	Nagpur Maharastra	India	2020-04-09	06:20:10	2020-04-11	11:10:10	4	Aadhar card	935573490	505
17	10126	Pulakit Sam	plc no 4 Laxmi nag	Goa Maharastra	India	2020-04-05	07:15:10	2020-04-07	11:10:10	2	Passport	733673590	401
18	10117	Kishore Kum	Mannat Gandhi naga	Rajkot Gujarat	India	2020-04-09	06:20:10	2020-04-13	11:10:10	5	Aadhar card	935673490	403
19	10113	Vaibhav Shi	Om Enclave Ulas na	Kanpur UP	India	2020-03-25	10:10:10	2020-03-30	12:10:10	5	Aadhar	921240011	101
20	10112	Vaibhavi Apt	Prakash Enclave Pr	Nagpur Maharastra	India	2020-03-20	11:10:10	2020-03-24	11:10:10	2	Aadhar	560324118	103

```
select cust_name from customer_info where ID = 'Passport';
```

Output pane	
Data Output Explain Messages History	
	cust_name character(40)
1	Krishna Pandey
2	Sanjana Pandey
3	Vijay Kumar Pandey
4	Pulakit Samrat

```
select cust_name,cust_id from customer_info where room_ty = 'ACDB' and ID = 'Aadhar card';
```

Output pane		
Data Output Explain Messages History		
	cust_name character(40)	cust_id integer
1	Kishore Kumar	10117
2	Ashok Malviya	10118
3	Manik Kumar	10120

d) Customer payment record table

```
create table cust_pay_rec(
    receipt_no    varchar(8) NOT NULL,
    cust_id       int NOT NULL,
    amount        int NOT NULL,
    pay_mode      char(6) CHECK(pay_mode IN('Credit','Debit','Cheque','Cash')),
    pay_date      date NOT NULL,
    PRIMARY KEY(Receipt_no),
    FOREIGN KEY(cust_id) references customer_info(cust_id)
);

alter table cust_pay_rec drop column amount;
alter table cust_pay_rec add column amount numeric(4,0);
alter table cust_pay_rec alter column amount type numeric(6,0);
```

```
insert into cust_pay_rec values(1101201,'10112','Debit','2020-03-24',6000);
insert into cust_pay_rec values(1101202,'10114','Cash','2020-04-03',12000);
insert into cust_pay_rec values(1101203,'10115','Debit','2020-04-04',7000);
insert into cust_pay_rec values(1101204,'10117','Credit','2020-04-14',22500);
insert into cust_pay_rec values(1101205,'10118','Debit','2020-04-13',22500);
insert into cust_pay_rec values(1101206,'10121','Debit','2020-04-11',16000);
insert into cust_pay_rec values(1101207,'10119','Cash','2020-04-14',30000);
insert into cust_pay_rec values(1101208,'10126','Credit','2020-04-07',9000);
insert into cust_pay_rec values(1101209,'10130','Debit','2020-04-13',17500);
insert into cust_pay_rec values(1101301,'10127','Debit','2020-04-13',15000);
insert into cust_pay_rec values(1101302,'10111','Credit','2020-04-11',32000);
select * from cust_pay_rec;
```

Output pane					
Data Output Explain Messages History					
	receipt_no character varying(8)	cust_id integer	pay_mode character(6)	pay_date date	amount numeric(6,0)
1	1101201	10112	Debit	2020-03-24	6000
2	1101202	10114	Cash	2020-04-03	12000
3	1101203	10115	Debit	2020-04-04	7000
4	1101204	10117	Credit	2020-04-14	22500
5	1101205	10118	Debit	2020-04-13	22500
6	1101206	10121	Debit	2020-04-11	16000
7	1101207	10119	Cash	2020-04-14	30000
8	1101208	10126	Credit	2020-04-07	9000
9	1101209	10130	Debit	2020-04-13	17500
10	1101301	10127	Debit	2020-04-13	15000
11	1101302	10111	Credit	2020-04-11	32000

```
select cust_id from cust_pay_rec where pay_mode = 'Credit';
```

Output pane	
Data Output Explain Messages History	
	cust_id integer
1	10117
2	10126
3	10127

INNER JOIN of customer_info & room_t

select cust_id, cust_name, room_ty, room_tariff from customer_info inner join room_t
on room_ty = room_type;

Data Output	Explain	Messages	History	
	cust_id integer	cust_name character(40)	room_ty character(4)	room_tariff numeric(4,0)
1	10129	Pushpa Agarwal	NASB	3000
2	10127	Deepen Chaturvedi	NASB	3000
3	10125	Kartar Singh	NASB	3000
4	10113	Vaibhav Shukla	NASB	3000
5	10112	Vaibhavi Apte	NASB	3000
6	10122	Anuj Lulu	ACSB	3500
7	10121	Pratap Bansal	ACSB	3500
8	10114	Krishna Pandey	ACSB	3500
9	10130	Manisha Kumari	NADB	4000
10	10128	Mallika Malviya	NADB	4000
11	10123	Sidhardh Shukla	NADB	4000
12	10116	Sanjana Pandey	NADB	4000
13	10115	Sanjay Singh	NADB	4000
14	10126	Pulakit Samrat	ACDB	4500
15	10120	Manik Kumar	ACDB	4500
16	10118	Ashok Malviya	ACDB	4500
17	10117	Kishore Kumar	ACDB	4500
18	10111	Kanika Kapoor	MS	8000
19	10124	Vijay Kumar Pandey	JS	6000
20	10119	Kishore Agarwal	JS	6000

RIGHT JOIN of customer pay record and customer information tables

select ci.cust_id, ci.cust_name, cpr.receipt_no from cust_pay_rec as cpr right join
customer_info as ci on ci.cust_id = cpr.cust_id;

Data Output	Explain	Messages	History
	cust_id integer	cust_name character(40)	receipt_no character varying(8)
1	10112	Vaibhavi Apte	1101201
2	10114	Krishna Pandy	1101202
3	10115	Sanjay Singh	1101203
4	10117	Kishore Kumar	1101204
5	10118	Ashok Malviya	1101205
6	10121	Pratap Bansal	1101206
7	10119	Kishore Agarwal	1101207
8	10126	Pulakit Samrat	1101208
9	10130	Manisha Kumari	1101209
10	10127	Deepen Chaturvedi	1101301
11	10111	Kanika kapoor	1101302
12	10113	Vaibhav Shukla	
13	10125	Kartar Singh	
14	10128	Mallika Malviya	
15	10120	Manik Kumar	
16	10124	Vijay Kumar Pandey	
17	10123	Sidhardh Shukla	
18	10116	Sanjana Pandy	
19	10122	Anuj Lulu	
20	10129	Pushpa Agarwal	

FULL JOIN of room tariff and room information tables

```
select ri.room_no, ri.room_type, rt.room_des, ri.room_status, rt.room_tariff from  
room_info as ri full join room_t as rt on ri.room_type = rt.room_type;
```

Data Output		Explain	Messages	History		
	room_no integer	room_type character(20)	room_des character(20)		room_status character(2)	room_tariff numeric(4,0)
1	103	NASB	NON AC Single bed	E		3000
2	104	NASB	NON AC Single bed	E		3000
3	105	NASB	NON AC Single bed	E		3000
4	201	NADB	NON AC Double bed	E		4000
5	202	NADB	NON AC Double bed	E		4000
6	204	NADB	NON AC Double bed	E		4000
7	206	NADB	NON AC Double bed	E		4000
8	302	ACSB	AC Single bed	E		3500
9	303	ACSB	AC Single bed	E		3500
10	304	ACSB	AC Single bed	E		3500
11	305	ACSB	AC Single bed	E		3500
12	306	ACSB	AC Single bed	E		3500
13	401	ACDB	AC Double bed	E		4500
14	402	ACDB	AC Double bed	E		4500
15	403	ACDB	AC Double bed	E		4500
16	404	ACDB	AC Double bed	E		4500
17	406	ACDB	AC Double bed	E		4500
18	501	JS	Junior suite	E		6000
19	503	JS	Junior suite	E		6000
20	504	MS	Master suite	E		8000
21	505	MS	Master suite	E		8000
22	506	MS	Master suite	E		8000
23	101	NASB	NON AC Single bed	O		3000
24	102	NASB	NON AC Single bed	O		3000
25	106	NASB	NON AC Single bed	O		3000
25	106	NASB	NON AC Single bed	O		3000
26	203	NADB	NON AC Double bed	O		4000
27	205	NADB	NON AC Double bed	O		4000
28	301	ACSB	AC Single bed	O		3500
29	405	ACDB	AC Double bed	O		4500
30	502	JS	Junior suite	O		6000

SUB – QUERIES

-- Display id and name of customers who have paid room rent in cash.

```
select cust_id, cust_name from customer_info where cust_id in(select cust_id from  
cust_pay_rec where pay_mode = 'Cash');
```

	Data Output	Explain	Messages	History
	cust_id integer	cust_name character(40)		
1	10114	Krishna Pandy		
2	10119	Kishore Agarwal		

-- Count total number of rooms of each type.

```
select room_type, count(room_type) from room_info where room_type = any (select  
room_type from room_t) group by room_type;
```

	Data Output	Explain	Messages	History
	room_type character(20)	count bigint		
1	NADB	6		
2	JS	3		
3	ACSB	6		
4	MS	3		
5	ACDB	6		
6	NASB	6		

-- Display id, name and phone number of customer who have occupied non AC single bed room

```
select cust_id,cust_name, mobile_no from customer_info where room_number in
(select room_no from room_info where room_type =
(select room_type from room_t where room_type = 'NASB') and room_status = 'O');
```

Output pane			
Data Output Explain Messages History			
	cust_id integer	cust_name character(40)	mobile_no bigint
1	10113	Vaibhav Shukla	1047834777
2	10125	Kartar Singh	9835672491
3	10129	Pushpa Agarwal	9755673490

TRANSACTION COMMANDS

```
begin;
update customer_info set mobile_no = 8821240011 where cust_id = 10113;
update customer_info set mobile_no = 7560324118 where cust_id = 10112;
savepoint A;
update customer_info set mobile_no = 7566324118 where cust_id = 10112;
rollback to A;
```

Output pane													
Data Output Explain Messages History													
	cust_id integer	cust_name character(40)	address_1 character varying(30)	address_2 character varying(30)	address_3 character varying(30)	check_in_dt date	check_in_tm time without time zone	check_out_dt date	check_out_tm time without time zone	no_of_days integer	id character(15)	mobile_no bigint	room_ty character(4) in
14	10129	Pushpa Agarwal	defence colony kolk	Goa Goa	India	2020-04-09	06:30:10	2020-04-13	11:10:10	5	Aadhar card	9755673490	NASB
15	10130	Manisha Kume	Pratap Enclave Pra	Hyderabad Telangan	India	2020-04-09	06:20:10	2020-04-13	11:10:10	5	Aadhar card	8935673490	NADB
16	10111	Manika kapoor	Manaya roofs Alok	Nagpur Maharashtra	India	2020-04-08	06:20:10	2020-04-11	11:10:10	4	Aadhar card	9835573490	MS
17	10126	Pulakit Sampl	no 4 Laxmi nag	Goa Maharashtra	India	2020-04-05	07:15:10	2020-04-07	11:10:10	2	Passport	7733673590	ACDB
18	10117	Kishore Kume	Mannat Gandhi naga	Rajkot Gujarat	India	2020-04-09	06:20:10	2020-04-13	11:10:10	5	Aadhar card	8835673490	ACDB
19	10113	Vaibhav Shukla	Om Enclave Ulas na	Hangpur UP	India	2020-03-25	10:10:10	2020-03-30	12:10:10	5	Aadhar	8821240011	NASB
20	10112	Vaibhavi Apt	Prakash Enclave Pr	Nagpur Maharashtra	India	2020-03-20	11:10:10	2020-03-24	11:10:10	2	Aadhar	7560324118	NASB

```
begin;
update room_info set room_status = 'E' where room_no = 201;
savepoint D;
update room_info set room_status = 'O' where room_no = 202;
rollback to D;
```

Output pane			
Data Output Explain Messages History			
	room_no integer	room_type character(20)	room_status character(2)
23	102	NASB	O
24	106	NASB	O
25	203	NADB	O
26	205	NADB	O
27	301	ACSB	O
28	405	ACDB	O
29	502	JS	O
30	201	NADB	E

Module 2: Staff personnel

a) Staff information table

```
create table employee(  
    emp_id    varchar(8) NOT NULL,  
    emp_name  char(40) NOT NULL,  
    DOB       date,  
    desig     char(10),  
    DOJ       date NOT NULL,  
    basic_sal numeric(6,0) CHECK(basic_sal > 999),  
    Aadhar_no int UNIQUE,  
    bank_name char(15),  
    Account_no int NOT NULL,  
    PRIMARY KEY(emp_id)  
);  
  
alter table employee alter column desig type char(18);  
  
insert into employee values ('R1212','Praneeti Bajaj','1980-12-03','General Manager','2019-08-08','95000','3777456','SBI','23343434');  
insert into employee values ('R1213','Pranoy Roy','1978-07-05','Deputy Manager','2018-07-28','55000','3787456','SBI','22243434');  
insert into employee values ('R1214','Gunjan Agarwal','1976-12-07','H.K Manager','2011-06-18','35000','310456','SBI','23344434');  
insert into employee values ('R1215','Kanishka Bajpai','1985-02-09','Sr. Receptionist','2010-05-22','35000','3766456','SBI','53343434');  
insert into employee values ('R1216','Kishore Kumar','1983-11-07','Receptionist','2015-08-08','15000','3785256','SBI','43343434');  
insert into employee values ('R1217','Komal Sinha','1982-10-05','Supervisor','2016-08-18','20000','3779456','SBI','21343434');  
insert into employee values ('R1218','Deepen Garg','1964-06-08','Maint. Enigineer','2017-03-05','55500','3757456','SBI','22343434');  
insert into employee values ('R1219','Manoj Tiwari','1992-05-03','Maint. Supervisor','2019-02-06','25000','3767456','SBI','25343434');  
insert into employee values ('R1210','Pallavi Joshi','1990-12-30','Accountant','2010-09-09','30000','3774456','SBI','28343434');  
insert into employee values ('R1211','Chandan Bajaj','1995-11-23','Accounts Manager','2014-11-08','55000','3737456','SBI','27343434');  
insert into employee values ('R1312','Meena Bajaj','1980-12-03','Admin. Manager','2019-08-08','45000','3777656','SBI','23343434');  
insert into employee values ('R1313','Purushotam Roy','1978-07-05','Clerk','2018-07-28','15000','3787476','SBI','22243434');  
insert into employee values ('R1314','Keerti Agarwal','1976-12-07','Helper','2011-06-18','10000','310496','SBI','23344434');  
insert into employee values ('R1315','Kokila Bajpai','1985-02-09','Helper','2010-05-22','12000','3766446','SBI','53343434');  
insert into employee values ('R1316','Ambal Kumar','1983-11-07','Receptionist','2015-08-
```

```

08','15000','3785236','SBI','43343434');
insert into employee values ('R1317','Amar Sinha','1982-10-05','Supervisor','2016-08-
18','20000','3719456','SBI','21343434');
insert into employee values ('R1318','Deepak Garg','1964-06-08','Maint. helper','2017-03-
05','15500','3757406','SBI','22343434');
insert into employee values ('R1319','Manisha Tiwari','1992-05-03','H.K. Supervisor','2019-02-
06','18000','3067456','SBI','25343434');
insert into employee values ('R1310','Pandu Joshi','1990-12-30','Worker','2010-09-
09','13000','3774416','SBI','28343434');
insert into employee values ('R1311','Charna Bajaj','1995-11-23','Accounts Clerk','2014-11-
08','25000','3633456','SBI','27343434');

```

```
SELECT * FROM employee;
```

Output pane									
Data Output Explain Messages History									
	e_id character varying(8)	name character(40)	dob date	desig character(18)	doj date	basic_sal numeric(6,0)	aadhar_no integer	bank_name character(15)	account_no integer
1	R1212	Praneeti Bajaj	1980-12-03	General Manager	2019-08-08	95000	3777456	SBI	23343434
2	R1213	Pranoy Roy	1978-07-05	Deputy Manager	2018-07-28	55000	3787456	SBI	22243434
3	R1214	Gunjan Agarwal	1976-12-07	H.K Manager	2011-06-18	35000	310456	SBI	23344434
4	R1215	Kanishka Bajpai	1985-02-09	Sr. Receptionist	2010-05-22	35000	3766456	SBI	53343434
5	R1216	Kishore Kumar	1983-11-07	Receptionist	2015-08-08	15000	3785256	SBI	43343434
6	R1217	Komal Sinha	1982-10-05	Supervisor	2016-08-18	20000	3779456	SBI	21343434
7	R1218	Deepen Garg	1964-06-08	Maint. Engineer	2017-03-05	55500	3757456	SBI	22343434
8	R1219	Manoj Tiwari	1992-05-03	Maint. Supervisor	2019-02-06	25000	3767456	SBI	25343434
9	R1210	Pallavi Joshi	1990-12-30	Accountant	2010-09-09	30000	3774456	SBI	28343434
10	R1211	Chandan Bajaj	1995-11-23	Accounts Manager	2014-11-08	55000	3737456	SBI	27343434
11	R1312	Meena Bajaj	1980-12-03	Admin. Manager	2019-08-08	45000	3777656	SBI	23343434
12	R1313	Purushotam Roy	1978-07-05	Clerk	2018-07-28	15000	3787476	SBI	22243434
13	R1314	Keerti Agarwal	1976-12-07	Helper	2011-06-18	10000	310496	SBI	23344434
14	R1315	Kokila Bajpai	1985-02-09	Helper	2010-05-22	12000	3766446	SBI	53343434
15	R1316	Ambal Kumar	1983-11-07	Receptionist	2015-08-08	15000	3785236	SBI	43343434
16	R1317	Amar Sinha	1982-10-05	Supervisor	2016-08-18	20000	3719456	SBI	21343434
17	R1318	Deepak Garg	1964-06-08	Maint. helper	2017-03-05	15500	3757406	SBI	22343434
18	R1319	Manisha Tiwari	1992-05-03	H.K. Supervisor	2019-02-06	18000	3067456	SBI	25343434
19	R1310	Pandu Joshi	1990-12-30	Worker	2010-09-09	13000	3774416	SBI	28343434
20	R1311	Charna Bajaj	1995-11-23	Accounts Clerk	2014-11-08	25000	3633456	SBI	27343434
21	R1410	Reema Shaha	1983-10-16	Accounts Clerk	2018-08-08	25000	3357890	SBI	24590118
22	R1411	Shashikal Raut	1995-10-16	Helper	2018-05-07	12000	3657810	SBI	14500118

```
select round( avg(basic_sal), 2) from employee;
```

Output pane	
Data Output Explain Messages History	
	round numeric
1	30450.00

```
select name, basic_sal, DOJ, DOB from employee where basic_sal > 20000;
```

Output pane				
Data Output Explain Messages History				
	name character(40)	basic_sal numeric(6,0)	doj date	dob date
1	Praneeti Bajaj	95000	2019-08-08	1980-12-03
2	Pranoy Roy	55000	2018-07-28	1978-07-05
3	Gunjan Agarwal	35000	2011-06-18	1976-12-07
4	Kanishka Bajpai	35000	2010-05-22	1985-02-09
5	Deepen Garg	55500	2017-03-05	1964-06-08
6	Manoj Tiwari	25000	2019-02-06	1992-05-03
7	Pallavi Joshi	30000	2010-09-09	1990-12-30
8	Chandan Bajaj	55000	2014-11-08	1995-11-23
9	Meena Bajaj	45000	2019-08-08	1980-12-03
10	Charna Bajaj	25000	2014-11-08	1995-11-23

select e_id, name, desig from employee where desig = 'Helper' or desig = 'Receptionist';

Output pane

Data Output	Explain	Messages	History
	e_id character varying(8)	name character(40)	desig character(18)
1	R1216	Kishore Kumar	Receptionist
2	R1314	Keerti Agarwal	Helper
3	R1315	Kokila Bajpai	Helper
4	R1316	Ambal Kumar	Receptionist

b) [Employee leaves table](#)

```
create table employee_leaves(
    emp_id    varchar(8) NOT NULL,
    leave_type char(8) CHECK(Leave_type IN('Medical','Casual','Earn')),
    from_dt   date NOT NULL,
    to_dt     date NOT NULL,
    no_of_days int,
    join_dt   date NOT NULL,
    last_leave_date date NOT NULL,
    FOREIGN KEY(emp_id)references employee(emp_id)
);
```

```
ALTER TABLE customer_info ADD Pay_amt    numeric(4,0);
alter table customer_info drop Pay_amt;
```

```
insert into employee_leaves values ('R1312','Casual','2020-03-19','2020-03-20','2','2020-03-21','2020-01-15');
insert into employee_leaves values ('R1319','Casual','2020-03-11','2020-03-13','3','2020-03-14','2020-02-15');
insert into employee_leaves values ('R1310','Casual','2020-06-19','2020-06-19','1','2020-05-21','2020-04-25');
insert into employee_leaves values ('R1315','Casual','2020-03-19','2020-03-20','1','2020-03-21','2020-04-16');
insert into employee_leaves values ('R1212','Medical','2020-07-15','2020-07-20','6','2020-03-21','2020-01-25');
insert into employee_leaves values ('R1215','Casual','2020-02-17','2020-02-17','1','2020-02-18','2020-02-02');
insert into employee_leaves values ('R1311','Casual','2020-03-11','2020-03-11','1','2020-03-12','2020-02-25');
insert into employee_leaves values ('R1312','Medical','2020-05-19','2020-05-22','4','2020-05-23','2020-01-15');
insert into employee_leaves values ('R1310','Casual','2020-06-19','2020-06-20','2','2020-06-21','2020-03-15');
insert into employee_leaves values ('R1214','Casual','2020-05-11','2020-05-11','1','2020-05-12','2020-04-15');
insert into employee_leaves values ('R1317','Medical','2020-06-19','2020-06-28','10','2020-06-29','2020-03-16');
insert into employee_leaves values ('R1216','Casual','2020-04-19','2020-04-20','2','2020-04-
```

```

21,'2020-01-19');
insert into employee_leaves values ('R1219','Earn','2020-05-10','2020-05-20','10','2020-05-
21','2020-01-27');
insert into employee_leaves values ('R1319','Earn','2020-07-02','2020-07-12','11','2020-07-
13','2020-04-03');
insert into employee_leaves values ('R1310','Casual','2020-02-09','2020-02-09','1','2020-02-
10','2020-01-06');

select * from employee_leaves;

```

Output pane							
	e_id character varying(8)	leave_type character(8)	from_dt date	to_dt date	no_of_days integer	join_dt date	last_leave_date date
1	R1312	Casual	2020-03-19	2020-03-20	2	2020-03-21	2020-01-15
2	R1319	Casual	2020-03-11	2020-03-13	3	2020-03-14	2020-02-15
3	R1310	Casual	2020-06-19	2020-06-19	1	2020-05-21	2020-04-25
4	R1315	Casual	2020-03-19	2020-03-20	1	2020-03-21	2020-04-16
5	R1212	Medical	2020-07-15	2020-07-20	6	2020-03-21	2020-01-25
6	R1215	Casual	2020-02-17	2020-02-17	1	2020-02-18	2020-02-02
7	R1311	Casual	2020-03-11	2020-03-11	1	2020-03-12	2020-02-25
8	R1312	Medical	2020-05-19	2020-05-22	4	2020-05-23	2020-01-15
9	R1310	Casual	2020-06-19	2020-06-20	2	2020-06-21	2020-03-15
10	R1214	Casual	2020-05-11	2020-05-11	1	2020-05-12	2020-04-15
11	R1317	Medical	2020-06-19	2020-06-28	10	2020-06-29	2020-03-16
12	R1216	Casual	2020-04-19	2020-04-20	2	2020-04-21	2020-01-19
13	R1219	Earn	2020-05-10	2020-05-20	10	2020-05-21	2020-01-27
14	R1319	Earn	2020-07-02	2020-07-12	11	2020-07-13	2020-04-03
15	R1310	Casual	2020-02-09	2020-02-09	1	2020-02-10	2020-01-06

```

select e_id from employee_leaves where leave_type = 'Casual';

```

Output pane	
	e_id character varying(8)
1	R1312
2	R1319
3	R1310
4	R1315
5	R1215
6	R1311
7	R1310
8	R1214
9	R1216
10	R1310

```

select e_id, sum(no_of_days) from employee_leaves group by e_id;

```

Output pane		
	e_id character varying(8)	sum bigint
1	R1319	14
2	R1216	2
3	R1317	10
4	R1311	1
5	R1312	6
6	R1315	1
7	R1215	1
8	R1219	10
9	R1212	6
10	R1310	4
11	R1214	1

LEFT JOIN of employee and employee leaves table

select e.e_id, e.name, el.from_dt, el.to_dt from employee as e left join employee_leaves as el on e.e_id = el.e_id;

Data Output	Explain	Messages	History	
	e_id character varying(8)	name character(40)	from_dt date	to_dt date
1	R1312	Meena Bajaj	2020-03-19	2020-03-20
2	R1319	Manisha Tiwari	2020-03-11	2020-03-13
3	R1310	Pandu Joshi	2020-06-19	2020-06-19
4	R1315	Kokila Bajpai	2020-03-19	2020-03-20
5	R1212	Praneeti Bajaj	2020-07-15	2020-07-20
6	R1215	Kanishka Bajpai	2020-02-17	2020-02-17
7	R1311	Charna Bajaj	2020-03-11	2020-03-11
8	R1312	Meena Bajaj	2020-05-19	2020-05-22
9	R1310	Pandu Joshi	2020-06-19	2020-06-20
10	R1214	Gunjan Agarwal	2020-05-11	2020-05-11
11	R1317	Amar Sinha	2020-06-19	2020-06-28
12	R1216	Kishore Kumar	2020-04-19	2020-04-20
13	R1219	Manoj Tiwari	2020-05-10	2020-05-20
14	R1319	Manisha Tiwari	2020-07-02	2020-07-12
15	R1310	Pandu Joshi	2020-02-09	2020-02-09
16	R1318	Deepak Garg		
17	R1213	Franoy Roy		
18	R1316	Ambal Kumar		
19	R1217	Komal Sinha		
20	R1218	Deepen Garg		
21	R1210	Pallavi Joshi		
22	R1211	Chandan Bajaj		
23	R1314	Keerti Agarwal		
24	R1313	Purushotam Roy		

NATURAL JOIN of employee and employee leaves tables

select e_id, name, desig, leave_type, last_leave_date from employee natural join employee_leaves;

Data Output	Explain	Messages	History		
	e_id character varying(8)	name character(40)	desig character(18)	leave_type character(8)	last_leave_date date
1	R1312	Meena Bajaj	Admin. Manager	Casual	2020-01-15
2	R1319	Manisha Tiwari	H.K. Supervisor	Casual	2020-02-15
3	R1310	Pandu Joshi	Worker	Casual	2020-04-25
4	R1315	Kokila Bajpai	Helper	Casual	2020-04-16
5	R1212	Praneeti Bajaj	General Manager	Medical	2020-01-25
6	R1215	Kanishka Bajpai	Sr. Receptionist	Casual	2020-02-02
7	R1311	Charna Bajaj	Accounts Clerk	Casual	2020-02-25
8	R1312	Meena Bajaj	Admin. Manager	Medical	2020-01-15
9	R1310	Pandu Joshi	Worker	Casual	2020-03-15
10	R1214	Gunjan Agarwal	H.K Manager	Casual	2020-04-15
11	R1317	Amar Sinha	Supervisor	Medical	2020-03-16
12	R1216	Kishore Kumar	Receptionist	Casual	2020-01-19
13	R1219	Manoj Tiwari	Maint. Supervisor	Earn	2020-01-27
14	R1319	Manisha Tiwari	H.K. Supervisor	Earn	2020-04-03
15	R1310	Pandu Joshi	Worker	Casual	2020-01-06

SUB - QUERIES

-- Display ID, name, designation of employee took leave for more than 3 days

select e_id, name, desig from employee where e_id in (select e_id from employee_leaves where no_of_days > 3);

Output pane			
Data Output	Explain	Messages	History
	e_id character varying(8)	name character(40)	desig character(18)
1	R1212	Praneeti Bajaj	General Manager
2	R1219	Manoj Tiwari	Maint. Supervisor
3	R1312	Meena Bajaj	Admin. Manager
4	R1317	Amar Sinha	Supervisor
5	R1319	Manisha Tiwari	H.K. Supervisor

-- Display ID, no_of_leaves of employee having salary less than 20000

```
select e_id, no_of_days from employee_leaves where e_id in(select e_id from employee
where basic_sal < 20000);
```

Output pane		
Data Output	Explain	Messages
	e_id character varying(8)	no_of_days integer
1	R1319	3
2	R1310	1
3	R1315	1
4	R1310	2
5	R1216	2
6	R1319	11
7	R1310	1

TRANSACTION COMMANDS

```
begin;
```

```
insert into employee values('R1410','Reema Shaha','1983-10-16','Accounts
Clerk','2018-08-08','25000','3357890','SBI','24590118');
```

```
insert into employee values('R1411','Shashikal Raut','1995-10-16','Helper','2018-05-
07','12000','3657810','SBI','14500118');
```

```
commit;
```

```
select * from employee;
```

Output pane									
Data Output	Explain	Messages	History						
	e_id character varying(8)	name character(40)	dob date	desig character(18)	doj date	basic_sal numeric(6,0)	aadhar_no integer	bank_name character(15)	account_no integer
16	R1317	Amar Sinha	1982	Supervisor	2016	20000	3719456	SBI	21343434
17	R1318	Deepak Garg	1964	Maint. help	2017	15500	3757406	SBI	22343434
18	R1319	Manisha Tiw	1992	H.K. Superv	2019	18000	3067456	SBI	25343434
19	R1310	Pandu Joshi	1990	Worker	2010	13000	3774416	SBI	28343434
20	R1311	Charna Baja	1995	Accounts Cle	2014	25000	3633456	SBI	27343434
21	R1410	Reema Shaha	1983	Accounts Cle	2018	25000	3357890	SBI	24590118
22	R1411	Shashikal R	1995	Helper	2018	12000	3657810	SBI	14500118

Module 3: Store module

a) Stock item details table

```
create table stock_item_details
(
  item_name char(30) unique,
  item_code int not null unique,
  item_type varchar(10) not null,
  item_value int unique not null,
  item_quantity int,
  item_uq char(3),
  item_rate int not null,
  item_last_date_of_buy date not null,
  min_stock int not null,
  primary key(item_code)
);
```

```
alter table stock_item_details alter column item_name type char(20);
alter table stock_item_details drop item_uq;
alter table stock_item_details alter column item_type type char(25);
alter table stock_item_details drop item_value;
alter table stock_item_details add item_value int;
```

```
insert into
stock_item_details(item_name,item_code,item_type,item_value,item_quantity,item_rate,item_last
_date_of_buy,min_stock) values('Toilet Paper',12500655,'Sanitary',3000,100,30,'2020-01-01',30);
insert into
stock_item_details(item_name,item_code,item_type,item_value,item_quantity,item_rate,item_last
_date_of_buy,min_stock) values('Facewash',89627804,'Complementary',500,100,5,'2020-01-
01',30);
insert into
stock_item_details(item_name,item_code,item_type,item_value,item_quantity,item_rate,item_last
_date_of_buy,min_stock) values('Shampoo',34871009,'Complementary',500,100,5,'2020-01-
02',30);
insert into
stock_item_details(item_name,item_code,item_type,item_value,item_quantity,item_rate,item_last
_date_of_buy,min_stock) values('Toilet Brush',87540094,'Cleaner',2700,90,30,'2020-01-02',50);
insert into
stock_item_details(item_name,item_code,item_type,item_value,item_quantity,item_rate,item_last
_date_of_buy,min_stock) values('Bed sheet',33994477,'Linen',40500,45,900,'2020-01-03',42);
insert into
stock_item_details(item_name,item_code,item_type,item_value,item_quantity,item_rate,item_last
_date_of_buy,min_stock) values('Detergent',87382933,'Laundry',4000,100,40,'2020-01-03',70);
insert into
stock_item_details(item_name,item_code,item_type,item_value,item_quantity,item_rate,item_last
```

```

_date_of_buy,min_stock) values('Dusting Cloth',33882970,'Cleaning',600,30,20,'2020-01-04',10);
insert into
stock_item_details(item_name,item_code,item_type,item_value,item_quantity,item_rate,item_last
_date_of_buy,min_stock) values('Mops',67386283,'Cleaning',1200,30,40,'2020-01-04',10);
insert into
stock_item_details(item_name,item_code,item_type,item_value,item_quantity,item_rate,item_last
_date_of_buy,min_stock) values('Toilet cleaner',27382700,'Cleaning',8000,100,80,'2020-01-
05',50);
insert into
stock_item_details(item_name,item_code,item_type,item_value,item_quantity,item_rate,item_last
_date_of_buy,min_stock) values('Soap',46800229,'Complementary',4000,100,40,'2020-01-05',50);
insert into
stock_item_details(item_name,item_code,item_type,item_value,item_quantity,item_rate,item_last
_date_of_buy,min_stock)values('Water jug',63598003,'Crockery',900,30,30,'2020-01-06',10);
insert into
stock_item_details(item_name,item_code,item_type,item_value,item_quantity,item_rate,item_last
_date_of_buy,min_stock)values('Trays',577883300,'Daily use',200,10,20,'2020-01-06',6);
insert into
stock_item_details(item_name,item_code,item_type,item_value,item_quantity,item_rate,item_last
_date_of_buy,min_stock)values('Towels',68889228,'Linen',4000,45,100,'2020-01-07',40);
insert into
stock_item_details(item_name,item_code,item_type,item_value,item_quantity,item_rate,item_last
_date_of_buy,min_stock)values('Blankets',70282987,'Linen',20000,45,500,'2020-01-07',40);
insert into
stock_item_details(item_name,item_code,item_type,item_value,item_quantity,item_rate,item_last
_date_of_buy,min_stock)values('Handwash',44488003,'Sanitary',10000,100,100,'2020-01-07',50);
insert into
stock_item_details(item_name,item_code,item_type,item_value,item_quantity,item_rate,item_last
_date_of_buy,min_stock)values('Glasses',43588002,'Crockery',6000,60,100,'2020-01-07',40);

select * from stock_item_details;

```

Output pane								
Data Output Explain Messages History								
	item_name character(20)	item_code integer	item_type character(25)	item_quantity integer	item_rate integer	item_last_date_of_buy date	min_stock integer	item_value integer
1	Toilet Paper	12500655	Sanitary	100	30	2020-01-01	30	3000
2	Facewash	89627804	Complementary	100	5	2020-01-01	30	500
3	Shampoo	34871009	Complementary	100	5	2020-01-02	30	500
4	Toilet Brush	87540094	Cleaner	90	30	2020-01-02	50	2700
5	Bed sheet	33994477	Linen	45	900	2020-01-03	42	40500
6	Detergent	87382933	Laundry	100	40	2020-01-03	70	4000
7	Dusting Cloth	33882970	Cleaning	30	20	2020-01-04	10	600
8	Mops	67386283	Cleaning	30	40	2020-01-04	10	1200
9	Toilet cleaner	27382700	Cleaning	100	80	2020-01-05	50	8000
10	Soap	46800229	Complementary	100	40	2020-01-05	50	4000
11	Water jug	63598003	Crockery	30	30	2020-01-06	10	900
12	Trays	577883300	Daily use	10	20	2020-01-06	6	200
13	Towels	68889228	Linen	45	100	2020-01-07	40	4000
14	Blankets	70282987	Linen	45	500	2020-01-07	40	20000
15	Handwash	44488003	Sanitary	100	100	2020-01-07	50	10000
16	Glasses	43588002	Crockery	60	100	2020-01-07	40	6000
17	Sanitizer	37448267	Complementar	100	40	2020-01-08	20	4000

select item_name , item_rate from stock_item_details where item_code= 63598003;

Output pane			
Data Output Explain Messages History			
	item_name character(20)	item_rate integer	
1	Water jug	30	

select count(item_name)from stock_item_details where min_stock between 10 and 50;

Output pane		
Data Output Explain Messages History		
	count bigint	
1	14	

SUB – QUERIES

--display the rate of all items which are greater than the rate of item_code=87382933.

select sid.item_code, sid.item_name, p.item_cost from stock_item_details as sid,
purchase as p where sid.item_code = p.item_code and p.item_cost >
(select distinct item_cost from purchase where item_code = 87382933);

Output pane			
Data Output Explain Messages History			
	item_code integer	item_name character(20)	item_cost integer
1	12500655	Toilet Paper	2100
2	46800229	Soap	2000
3	12500655	Toilet Paper	2100
4	12500655	Toilet Paper	2100
5	46800229	Soap	2000
6	12500655	Toilet Paper	2100

--display name of items whose quantity is less than 10

select item_name, item_code from stock_item_details where item_code in(select
item_code from purchase where quantity < 10);

Output pane		
Data Output Explain Messages History		
	item_name character(20)	item_code integer
1	Trays	577883300

TRANSACTION COMMANDS

```
begin;  
insert into  
stock_item_details(item_name,item_code,item_type,item_value,item_quantity,item_ra  
te,item_last_date_of_buy,min_stock)  
values('Toothpaste',37448267,'Complementary',4000,100,40,'2020-01-08',20);  
savepoint E;  
select * from stock_item_details;
```

Output pane

	item_name character(20)	item_code integer	item_type character(25)	item_quantity integer	item_rate integer	item_last_date_of_buy date	min_stock integer	item_value integer
11	Water jug	63598003	Crockery	30	30	2020-01-06	10	900
12	Trays	577883300	Daily use	10	20	2020-01-06	6	200
13	Towels	68889228	Linen	45	100	2020-01-07	40	4000
14	Blankets	70282987	Linen	45	500	2020-01-07	40	20000
15	Handwash	44488003	Sanitary	100	100	2020-01-07	50	10000
16	Glasses	43588002	Crockery	60	100	2020-01-07	40	6000
17	Toothpaste	37448267	Complementar	100	40	2020-01-08	20	4000

```
update stock_item_details set item_name = 'Sanitizer' where item_code = 37448267;
```

Output pane

	item_name character(20)	item_code integer	item_type character(25)	item_quantity integer	item_rate integer	item_last_date_of_buy date	min_stock integer	item_value integer
11	Water jug	63598003	Crockery	30	30	2020-01-06	10	900
12	Trays	577883300	Daily use	10	20	2020-01-06	6	200
13	Towels	68889228	Linen	45	100	2020-01-07	40	4000
14	Blankets	70282987	Linen	45	500	2020-01-07	40	20000
15	Handwash	44488003	Sanitary	100	100	2020-01-07	50	10000
16	Glasses	43588002	Crockery	60	100	2020-01-07	40	6000
17	Sanitizer	37448267	Complementar	100	40	2020-01-08	20	4000

```
rollback to E;  
select * from stock_item_details;  
delete from stock_item_details where item_code = 37448267;
```

Output pane

	item_name character(20)	item_code integer	item_type character(25)	item_quantity integer	item_rate integer	item_last_date_of_buy date	min_stock integer	item_value integer
11	Water jug	63598003	Crockery	30	30	2020-01-06	10	900
12	Trays	577883300	Daily use	10	20	2020-01-06	6	200
13	Towels	68889228	Linen	45	100	2020-01-07	40	4000
14	Blankets	70282987	Linen	45	500	2020-01-07	40	20000
15	Handwash	44488003	Sanitary	100	100	2020-01-07	50	10000
16	Glasses	43588002	Crockery	60	100	2020-01-07	40	6000
17	Toothpaste	37448267	Complementar	100	40	2020-01-08	20	4000

b) Capital inventory table

```
create table capital_inventory  
(  
inventory_number int not null unique,  
inv_name char not null,
```

```

brand_name char not null,
installment_date date not null,
maintainance_duration varchar not null,
primary key(inventory_number)
);
alter table capital_inventory alter column inv_name type char(20);
alter table capital_inventory alter column brand_name type char(20);
alter table capital_inventory alter column maintainance_duration type varchar(7);
alter table capital_inventory rename column maintainance_duration to maintenance_duration;
alter table capital_inventory alter column maintenance_duration type varchar(10);

insert into capital_inventory values(73873877,'AC','Samsung','2018-04-05','1 year');
insert into capital_inventory values(78634783,'Bed','Gordrej','2019-07-09','null');
insert into capital_inventory values(35657534,'Chair','Neelkamal','2018-07-11','null');
insert into capital_inventory values(67837463,'TV','Panasonic','2018-07-12','1 year');
insert into capital_inventory values(86643467,'Fan','USHA','2018-07-13','1 year');
insert into capital_inventory values(74837483,'Tube light','Syska','2020-07-15','1 year');
insert into capital_inventory values(53667742,'Washing machine','Whirlpool','2018-07-17','6
months');
insert into capital_inventory values(80755008,'Computers','Acer','2018-07-19','1 year');
insert into capital_inventory values(56634567,'Telephone','Verizon','2020-07-19','6 months');
insert into capital_inventory values(64836884,'Water Geyser','Bajaj','2019-07-20','6 months');
insert into capital_inventory values(53735727,'Cupboard','Gordrej','2020-07-21','null');
insert into capital_inventory values(75687890,'Table','Gordrej','2020-07-21','null');
insert into capital_inventory values(83749477,'Refrigerator','Samsung','2018-07-22','6 months');
insert into capital_inventory values(58979708,'Sofa','Comfort','2020-07-22','null');
insert into capital_inventory values(57897890,'Bed lamp','Philips','2018-07-23','6 months');
insert into capital_inventory values(89088760,'Mattress','Gordrej','2020-07-23','null');
insert into capital_inventory values(98897778,'Pillow','Sleepwell','2020-07-24','null');
insert into capital_inventory values(69977088,'Clock','Ajanta','20218-07-24','3 month');
insert into capital_inventory values(46746744,'Fire extinguisher','Safeu','2020-07-26','1 year');
insert into capital_inventory values(77689799,'Generator','Kirloskar','2018-07-28','6 months');
insert into capital_inventory values(48440001,'Water pump','Kirloskar','2020-06-15','6 months');
insert into capital_inventory values(34562901,'Sewing machine','USHA','2018-03-14','1 month');
insert into capital_inventory values(34552802,'Bed side table','Gordrej','2018-03-14','1 month');

select * from capital_inventory;

```

Output pane					
Data Output Explain Messages History					
	inventory_number integer	inv_name character(20)	brand_name character(20)	installment_date date	maintenance_duration character varying(10)
1	73873877	AC	Samsung	2018-04-05	1 year
2	78634783	Bed	Gordrej	2019-07-09	null
3	35657534	Chair	Neelkamal	2018-07-11	null
4	67837463	TV	Panasonic	2018-07-12	1 year
5	86643467	Fan	USHA	2018-07-13	1 year
6	74837483	Tube light	Syska	2020-07-15	1 year
7	53667742	Washing machi	Whirlpool	2018-07-17	6 months
8	80755008	Computers	Acer	2018-07-19	1 year
9	56634567	Telephone	Verizon	2020-07-19	6 months
10	64836884	Water Geyser	Bajaj	2019-07-20	6 months
11	53735727	Cupboard	Gordrej	2020-07-21	null
12	75687890	Table	Gordrej	2020-07-21	null
13	83749477	Refrigerator	Samsung	2018-07-22	6 months
14	58979708	Sofa	Comfort	2020-07-22	null
15	57897890	Bed lamp	Philips	2018-07-23	6 months
16	89088760	Mattress	Gordrej	2020-07-23	null
17	98897778	Pillow	Sleepwell	2020-07-24	null
18	69977088	Clock	Ajanta	20218-07-24	3 month
19	46746744	Fire extinguish	Safeu	2020-07-26	1 year
20	77689799	Generator	Kirloskar	2018-07-28	6 months
21	48440001	Water pump	Kirloskar	2020-06-15	6 months
22	34562901	Sewing machine	USHA	2018-03-14	1 month
23	34552802	Bed side table	Gordrej	2018-03-14	1 month

select count(inv_name) from capital_inventory where brand_name = 'Bajaj' and
inv_name not like 'A%';

Output pane	
Data Output Explain Messages History	
	count bigint
1	1

select inv_name,brand_name from capital_inventory where maintenance_duration = '6
months' or maintenance_duration = 'null';

Output pane		
Data Output Explain Messages History		
	inv_name character(20)	brand_name character(20)
1	Bed	Gordrej
2	Chair	Neelkamal
3	Washing machine	Whirlpool
4	Telephone	Verizon
5	Water Geyser	Bajaj
6	Cupboard	Gordrej
7	Table	Gordrej
8	Refrigerator	Samsung
9	Sofa	Comfort
10	Bed lamp	Philips
11	Mattress	Gordrej
12	Pillow	Sleepwell
13	Generator	Kirloskar
14	Water pump	Kirloskar

TRANSACTION COMMANDS

```
begin;
delete from capital_inventory where inventory_number = 73873877 ;
commit;
```



```
select * from capital_inventory;
```

Output pane					
Data Output Explain Messages History					
	inventory_number integer	inv_name character(20)	brand_name character(20)	installment_date date	maintenance_duration character varying(10)
1	78634783	Bed	Gordrej	2019-07-09	null
2	35657534	Chair	Neelkamal	2018-07-11	null
3	67837463	TV	Panasonic	2018-07-12	1 year
4	86643467	Fan	USHA	2018-07-13	1 year
5	74837483	Tube light	Syska	2020-07-15	1 year
6	53667742	Washing mach	Whirlpool	2018-07-17	6 months
7	80755008	Computers	Acer	2018-07-19	1 year
8	56634567	Telephone	Verizon	2020-07-19	6 months

```
begin;
```

```
insert into capital_inventory values(87674673,'Heater','Philips','2018-04-07','1 year');
```

```
select * from capital_inventory;
```

Output pane					
Data Output Explain Messages History					
	inventory_number integer	inv_name character(20)	brand_name character(20)	installment_date date	maintenance_duration character varying(10)
17	69977088	Clock	Ajanta	20218-07-24	3 month
18	46746744	Fire extingui	Safeu	2020-07-26	1 year
19	77689799	Generator	Kirloskar	2018-07-28	6 months
20	48440001	Water pump	Kirloskar	2020-06-15	6 months
21	34562901	Sewing machi	USHA	2018-03-14	1 month
22	34552802	Bed side tak	Gordrej	2018-03-14	1 month
23	87674673	Heater	Philips	2018-04-07	1 year

```
update capital_inventory set inv_name = 'Night lamp' where inventory_number = 87674673 ;
```

```
select * from capital_inventory;
```

```
savepoint A;
```

Output pane					
Data Output Explain Messages History					
	inventory_number integer	inv_name character(20)	brand_name character(20)	installment_date date	maintenance_duration character varying(10)
17	69977088	Clock	Ajanta	20218-07-24	3 month
18	46746744	Fire extingui	Safeu	2020-07-26	1 year
19	77689799	Generator	Kirloskar	2018-07-28	6 months
20	48440001	Water pump	Kirloskar	2020-06-15	6 months
21	34562901	Sewing machi	USHA	2018-03-14	1 month
22	34552802	Bed side tak	Gordrej	2018-03-14	1 month
23	87674673	Night lamp	Philips	2018-04-07	1 year

```
insert into capital_inventory values(97473889,'Tubelight','philips','2018-06-09','6 months');
```

```
savepoint B;
```

Output pane					
Data Output Explain Messages History					
	inventory_number integer	inv_name character(20)	brand_name character(20)	installment_date date	maintenance_duration character varying(10)
18	46746744	Fire extingui	Safeu	2020-07-26	1 year
19	77689799	Generator	Kirloskar	2018-07-28	6 months
20	48440001	Water pump	Kirloskar	2020-06-15	6 months
21	34562901	Sewing machi	USHA	2018-03-14	1 month
22	34552802	Bed side tak	Gordrej	2018-03-14	1 month
23	87674673	Night lamp	Philips	2018-04-07	1 year
24	97473889	Tubelight	philips	2018-06-09	6 months

rollback to A;
select * from capital_inventory ;

Output pane					
Data Output Explain Messages History					
	inventory_number integer	inv_name character(20)	brand_name character(20)	installment_date date	maintenance_duration character varying(10)
17	69977088	Clock	Ajanta	20218-07-24	3 month
18	46746744	Fire extingui	Safeu	2020-07-26	1 year
19	77689799	Generator	Kirloskar	2018-07-28	6 months
20	48440001	Water pump	Kirloskar	2020-06-15	6 months
21	34562901	Sewing machi	USHA	2018-03-14	1 month
22	34552802	Bed side tak	Gordrej	2018-03-14	1 month
23	87674673	Night lamp	Philips	2018-04-07	1 year

Module 4:Maintenance module

```
create table maintenance
(
  inv_code   varchar(8) not null,
  breakdown_dt date not null,
  restoration_dt date not null,
  breakdown_type char(10) not null,
  breakdown_nature char(10),
  repair_charges int not null,
  primary key(inv_code)
);
alter table maintenance drop breakdown_nature;
alter table maintenance drop inv_code;
alter table maintenance add column inv_code int;
alter table maintenance add foreign key(inv_code) references
capital_inventory(inventory_number) on delete cascade on update cascade;
```

```
insert into maintenance
(inv_code,breakdown_dt,restoration_dt,breakdown_type,repair_charges)values('73873877','2020-
04-05','2020-04-08','Electrical',13000);
insert into maintenance
```

```

(inv_code,breakdown_dt,restoration_dt,breakdown_type,repair_charges)values('35657534','2020-04-02','2020-04-04','Electrical',3000);
insert into maintenance
(inv_code,breakdown_dt,restoration_dt,breakdown_type,repair_charges)values('86643467','2020-04-03','2020-04-04','Electrical',1000);
insert into maintenance
(inv_code,breakdown_dt,restoration_dt,breakdown_type,repair_charges)values('74837483','2020-04-02','2020-04-04','Electrical',1000);
insert into maintenance
(inv_code,breakdown_dt,restoration_dt,breakdown_type,repair_charges)values('53667742','2020-04-06','2020-04-06','Electrical',4000);
insert into maintenance
(inv_code,breakdown_dt,restoration_dt,breakdown_type,repair_charges)values('80755008','2020-05-02','2020-05-04','Electrical',15000);
insert into maintenance
(inv_code,breakdown_dt,restoration_dt,breakdown_type,repair_charges)values('64836884','2020-05-03','2020-05-04','Electrical',20000);
insert into maintenance
(inv_code,breakdown_dt,restoration_dt,breakdown_type,repair_charges)values('46746744','2020-05-04','2020-05-06','Serviceing',14000);
insert into maintenance
(inv_code,breakdown_dt,restoration_dt,breakdown_type,repair_charges)values('77689799','2020-05-07','2020-05-08','Electrical',33000);
insert into maintenance (inv_code,breakdown_dt,restoration_dt,breakdown_type,repair_charges)
values('34562901','2020-06-02','2020-06-04','Mechanical',2000);

```

```
select * from maintenance;
```

Output pane					
Data Output Explain Messages History					
	inv_code character varying(8)	breakdown_dt date	restoration_dt date	breakdown_type character(10)	repair_charges integer
1	73873877	2020-04-05	2020-04-08	Electrical	13000
2	35657534	2020-04-02	2020-04-04	Electrical	3000
3	86643467	2020-04-03	2020-04-04	Electrical	1000
4	74837483	2020-04-02	2020-04-04	Electrical	1000
5	53667742	2020-04-06	2020-04-06	Electrical	4000
6	80755008	2020-05-02	2020-05-04	Electrical	15000
7	64836884	2020-05-03	2020-05-04	Electrical	20000
8	46746744	2020-05-04	2020-05-06	Serviceing	14000
9	77689799	2020-05-07	2020-05-08	Electrical	33000
10	34562901	2020-06-02	2020-06-04	Mechanical	2000

select breakdown_dt, restoration_dt from maintenance where breakdown_type in ('Mechanical','Servicing');

Output pane		
Data Output	Explain	Messages
History		
	breakdown_dt date	restoration_dt date
1	2020-06-02	2020-06-04
2	2020-05-04	2020-05-06

LEFT JOIN of capital inventory and maintenance tables

select inv_name,brand_name,maintenance_duration,breakdown_type,repair_charges from capital_inventory left outer join maintenance on inventory_number=inv_code;

Data Output	Explain	Messages	History			
	inv_name character(20)	brand_name character(20)	maintenance_duration character varying(10)	breakdown_type character(10)	repair_charges integer	
1	AC	Samsung	1 year	Electrical	13000	
2	Chair	Neelkamal	null	Electrical	3000	
3	Fan	USHA	1 year	Electrical	1000	
4	Tube light	Syska	1 year	Electrical	1000	
5	Washing machine	Whirlpool	6 months	Electrical	4000	
6	Computers	Acer	1 year	Electrical	15000	
7	Water Geyser	Bajaj	6 months	Electrical	20000	
8	Generator	Kirloskar	6 months	Electrical	33000	
9	Sewing machine	USHA	1 month	Mechanical	2000	
10	Fire extinguisher	Safeu	1 year	Servicing	14000	
11	Mattress	Gordrej	null			
12	Cupboard	Gordrej	null			
13	TV	Panasonic	1 year			
14	Clock	Ajanta	3 month			
15	Water pump	Kirloskar	6 months			
16	Pillow	Sleepwell	null			
17	Telephone	Verizon	6 months			
18	Refrigerator	Samsung	6 months			
19	Table	Gordrej	null			
20	Bed lamp	Philips	6 months			
21	Sofa	Comfort	null			
22	Bed side table	Gordrej	1 month			
23	Bed	Gordrej	null			

SUB – QUERIES

--Display brand_name and installation date of item whose breakdown type is electrical

select brand_name,installment_date from capital_inventory where inventory_number in (select inv_code from maintenance where breakdown_type = 'Electrical');

Output pane			
Data Output	Explain	Messages	History
	inv_name character(20)	brand_name character(20)	installment_date date
1	Generator	Kirloskar	2018-07-28
2	Chair	Neelkamal	2018-07-11
3	Fan	USHA	2018-07-13
4	Washing mach	Whirlpool	2018-07-17
5	AC	Samsung	2018-04-05
6	Tube light	Syska	2020-07-15
7	Computers	Acer	2018-07-19
8	Water Geyser	Bajaj	2019-07-20


```
--display inventory number, inv_name ,brand_name, installment_date,
maintainance_duration where repair_charges are greater than or equal to 15000.
select * from capital_inventory where inventory_number in(select inv_code from
maintenance where repair_charges >= 15000);
```

Output pane					
Data Output Explain Messages History					
	inventory_number integer	inv_name character(20)	brand_name character(20)	installment_date date	maintenance_duration character varying(10)
1	80755008	Computers	Acer	2018-07-19	1 year
2	64836884	Water Geyser	Bajaj	2019-07-20	6 months
3	77689799	Generator	Kirloskar	2018-07-28	6 months

Module 5: Purchase module

```
create table purchase
```

```
(
order_number int not null unique,
date_of_buy date not null,
quantity int not null,
rate int not null,
item_code int unique not null,
delivery_date date not null,
order_status varchar,
primary key(order_number)
);
```

```
alter table purchase alter column order_status type varchar(8);
alter table purchase add check (order_status in ('open','close','paropen'));
alter table purchase rename column rate to cost;
alter table purchase rename column cost to item_cost;
alter table purchase drop item_code;
alter table purchase add item_code int;
alter table purchase drop order_number;
alter table purchase add order_number int;
alter table purchase alter column order_number set not null;
alter table purchase add foreign key(item_code)references
stock_item_details(item_code) on delete cascade on update cascade;
```

```
insert into
```

```
purchase(order_number,date_of_buy,quantity,item_cost,item_code,delivery_date,order_
status)values(2011030,'2020-03-03',70,2100,12500655,'2020-03-04','close');
```

```
insert into
```

```
purchase(order_number,date_of_buy,quantity,item_cost,item_code,delivery_date,order_
status)values(2011031,'2020-03-04',70,350,89627804,'2020-03-06','close');
```

```
insert into
```

```
purchase(order_number,date_of_buy,quantity,item_cost,item_code,delivery_date,order_
```

```

status)values(2011032,'2020-03-05',70,350,34871009,'2020-03-07','close');
insert into
purchase(order_number,date_of_buy,quantity,item_cost,item_code,delivery_date,order_
status)values(2011033,'2020-04-03',40,1200,87540094,'2020-04-05','close');
insert into
purchase(order_number,date_of_buy,quantity,item_cost,item_code,delivery_date,order_
status)values(2011034,'2020-04-05',30,1200,87382933,'2020-04-07','close');
insert into
purchase(order_number,date_of_buy,quantity,item_cost,item_code,delivery_date,order_
status)values(2011035,'2020-04-06',10,200,33882970,'2020-03-07','close');
insert into
purchase(order_number,date_of_buy,quantity,item_cost,item_code,delivery_date,order_
status)values(2011036,'2020-05-03',50,2000,46800229,'2020-05-04','close');
insert into
purchase(order_number,date_of_buy,quantity,item_cost,item_code,delivery_date,order_
status)values(2011037,'2020-05-03',4,80,577883300,'2020-05-04','close');
insert into
purchase(order_number,date_of_buy,quantity,item_cost,item_code,delivery_date,order_
status)values(2011038,'2020-05-04',70,2100,12500655,'2020-05-05','close');
insert into
purchase(order_number,date_of_buy,quantity,item_cost,item_code,delivery_date,order_
status)values(2011039,'2020-05-05',60,300,348710094,'2020-05-06','close');
insert into
purchase(order_number,date_of_buy,quantity,item_cost,item_code,delivery_date,order_
status)values(2011040,'2020-05-03',70,2100,12500655,'2020-05-04','close');
insert into
purchase(order_number,date_of_buy,quantity,item_cost,item_code,delivery_date,order_
status)values(2011041,'2020-05-04',70,350,89627804,'2020-05-06','close');
insert into
purchase(order_number,date_of_buy,quantity,item_cost,item_code,delivery_date,order_
status)values(2011042,'2020-05-05',70,350,34871009,'2020-05-07','close');
insert into
purchase(order_number,date_of_buy,quantity,item_cost,item_code,delivery_date,order_
status)values(2011043,'2020-06-03',40,1200,87540094,'2020-06-05','close');
insert into
purchase(order_number,date_of_buy,quantity,item_cost,item_code,delivery_date,order_
status)values(2011044,'2020-06-05',30,1200,87382933,'2020-06-07','close');
insert into
purchase(order_number,date_of_buy,quantity,item_cost,item_code,delivery_date,order_
status)values(2011045,'2020-06-06',10,200,33882970,'2020-06-07','close');
insert into
purchase(order_number,date_of_buy,quantity,item_cost,item_code,delivery_date,order_
status)values(2011046,'2020-06-03',50,2000,46800229,'2020-06-04','close');
insert into
purchase(order_number,date_of_buy,quantity,item_cost,item_code,delivery_date,order_
status)values(2011047,'2020-06-03',4,80,577883300,'2020-06-04','close');
insert into

```

```
purchase(order_number,date_of_buy,quantity,item_cost,item_code,delivery_date,order_
status)values(2011048,'2020-06-04',70,2100,12500655,'2020-06-05','close');
```

```
insert into
```

```
purchase(order_number,date_of_buy,quantity,item_cost,item_code,delivery_date,order_
status)values(2011049,'2020-06-05',60,300,348710094,'2020-06-06','close');
```

```
select * from purchase;
```

Output pane							
Data Output Explain Messages History							
	date_of_buy date	quantity integer	item_cost integer	delivery_date date	order_status character varying(8)	item_code integer	order_number integer
1	2020-03-03	70	2100	2020-03-04	close	12500655	2011030
2	2020-03-04	70	350	2020-03-06	close	89627804	2011031
3	2020-03-05	70	350	2020-03-07	close	34871009	2011032
4	2020-04-03	40	1200	2020-04-05	close	87540094	2011033
5	2020-04-05	30	1200	2020-04-07	close	87382933	2011034
6	2020-04-06	10	200	2020-03-07	close	33882970	2011035
7	2020-05-03	50	2000	2020-05-04	close	46800229	2011036
8	2020-05-03	4	80	2020-05-04	close	577883300	2011037
9	2020-05-04	70	2100	2020-05-05	close	12500655	2011038
10	2020-05-05	60	300	2020-05-06	close	348710094	2011039
11	2020-05-03	70	2100	2020-05-04	close	12500655	2011040
12	2020-05-04	70	350	2020-05-06	close	89627804	2011041
13	2020-05-05	70	350	2020-05-07	close	34871009	2011042
14	2020-06-03	40	1200	2020-06-05	close	87540094	2011043
15	2020-06-05	30	1200	2020-06-07	close	87382933	2011044
16	2020-06-06	10	200	2020-06-07	close	33882970	2011045
17	2020-06-03	50	2000	2020-06-04	close	46800229	2011046
18	2020-06-03	4	80	2020-06-04	close	577883300	2011047
19	2020-06-04	70	2100	2020-06-05	close	12500655	2011048
20	2020-06-05	60	300	2020-06-06	close	348710094	2011049

```
select order_number , item_cost from purchase where item_code = 87382933;
```

Output pane		
Data Output Explain Messages History		
	order_number integer	item_cost integer
1	2011034	1200
2	2011044	1200

```
select * from purchase where item_cost between 300 and 1200 and order_status =
'close';
```

Output pane							
Data Output Explain Messages History							
	date_of_buy date	quantity integer	item_cost integer	delivery_date date	order_status character varying(8)	item_code integer	order_number integer
1	2020-03-04	70	350	2020-03-06	close	89627804	2011031
2	2020-03-05	70	350	2020-03-07	close	34871009	2011032
3	2020-04-03	40	1200	2020-04-05	close	87540094	2011033
4	2020-04-05	30	1200	2020-04-07	close	87382933	2011034
5	2020-05-05	60	300	2020-05-06	close	348710094	2011039
6	2020-05-04	70	350	2020-05-06	close	89627804	2011041
7	2020-05-05	70	350	2020-05-07	close	34871009	2011042
8	2020-06-03	40	1200	2020-06-05	close	87540094	2011043
9	2020-06-05	30	1200	2020-06-07	close	87382933	2011044
10	2020-06-05	60	300	2020-06-06	close	348710094	2011049

INNER JOIN of Stock item details and purchase tables

select sid.item_name, sid.item_type, pur.order_number, pur.date_of_buy from
stock_item_details as sid inner join purchase as pur on sid.item_code=pur.item_code;

Data Output	Explain	Messages	History		
	item_name character(20)	item_type character(25)	order_number integer	date_of_buy date	
1	Toilet Paper	Sanitary	2011030	2020-03-03	
2	Facewash	Complementary	2011031	2020-03-04	
3	Shampoo	Complementary	2011032	2020-03-05	
4	Toilet Brush	Cleaner	2011033	2020-04-03	
5	Detergent	Laundry	2011034	2020-04-05	
6	Dusting Cloth	Cleaning	2011035	2020-04-06	
7	Soap	Complementary	2011036	2020-05-03	
8	Trays	Daily use	2011037	2020-05-03	
9	Toilet Paper	Sanitary	2011038	2020-05-04	
10	Toilet Paper	Sanitary	2011040	2020-05-03	
11	Facewash	Complementary	2011041	2020-05-04	
12	Shampoo	Complementary	2011042	2020-05-05	
13	Toilet Brush	Cleaner	2011043	2020-06-03	
14	Detergent	Laundry	2011044	2020-06-05	
15	Dusting Cloth	Cleaning	2011045	2020-06-06	
16	Soap	Complementary	2011046	2020-06-03	
17	Trays	Daily use	2011047	2020-06-03	
18	Toilet Paper	Sanitary	2011048	2020-06-04	

FULL JOIN of stock item details and purchase tables

select sid.item_name, sid.item_value, sid.item_quantity, pur.delivery_date ,
pur.order_status from stock_item_details as sid full outer join purchase as pur on
sid.item_code = pur.item_code;

Data Output	Explain	Messages	History			
	item_name character(20)	item_value integer	item_quantity integer	delivery_date date	order_status character varying(8)	
1	Toilet Paper	3000	100	2020-03-04	close	
2	Facewash	500	100	2020-03-06	close	
3	Shampoo	500	100	2020-03-07	close	
4	Toilet Brush	2700	90	2020-04-05	close	
5	Detergent	4000	100	2020-04-07	close	
6	Dusting Cloth	600	30	2020-03-07	close	
7	Soap	4000	100	2020-05-04	close	
8	Trays	200	10	2020-05-04	close	
9	Toilet Paper	3000	100	2020-05-05	close	
11	Toilet Paper	3000	100	2020-05-04	close	
12	Facewash	500	100	2020-05-06	close	
13	Shampoo	500	100	2020-05-07	close	
14	Toilet Brush	2700	90	2020-06-05	close	
15	Detergent	4000	100	2020-06-07	close	
16	Dusting Cloth	600	30	2020-06-07	close	
17	Soap	4000	100	2020-06-04	close	
18	Trays	200	10	2020-06-04	close	
19	Toilet Paper	3000	100	2020-06-05	close	
21	Handwash	10000	100			
22	Glasses	6000	60			
23	Towels	4000	45			
24	Water jug	900	30			
25	Toilet clear	8000	100			
26	Blankets	20000	45			

NATURAL JOIN of purchase and stock item details tables

select * from stock_item_details natural join purchase;

Data Output	Explain	Messages	History											
	item_code integer	item_name character(20)	item_type character(25)	item_quantity integer	item_rate integer	item_last_date_of_buy date	min_stock integer	item_value integer	date_of_buy date	quantity integer	item_cost integer	delivery_date date	order_status character varying(8)	order_number integer
1	12500655	Toilet Paper	Sanitary	100	30	2020-01-01	30	3000	2020-03-03	70	2100	2020-03-04	close	2011030
2	89627804	Facewash	Complementary	100	5	2020-01-01	30	500	2020-03-04	70	350	2020-03-06	close	2011031
3	34871009	Shampoo	Complementary	100	5	2020-01-02	30	500	2020-03-05	70	350	2020-03-07	close	2011032
4	87540094	Toilet Brush	Cleaner	90	30	2020-01-02	50	2700	2020-04-03	40	1200	2020-04-05	close	2011033
5	87382933	Detergent	Laundry	100	40	2020-01-03	70	4000	2020-04-05	30	1200	2020-04-07	close	2011034
6	33882970	Dusting Cloth	Cleaning	30	20	2020-01-04	10	600	2020-04-06	10	200	2020-03-07	close	2011035
7	46800229	Soap	Complementary	100	40	2020-01-05	50	4000	2020-05-03	50	2000	2020-05-04	close	2011036
8	577883300	Trays	Daily use	10	20	2020-01-06	6	200	2020-05-03	4	80	2020-05-04	close	2011037
9	12500655	Toilet Paper	Sanitary	100	30	2020-01-01	30	3000	2020-05-04	70	2100	2020-05-05	close	2011038
10	12500655	Toilet Paper	Sanitary	100	30	2020-01-01	30	3000	2020-05-03	70	2100	2020-05-04	close	2011040
11	89627804	Facewash	Complementary	100	5	2020-01-01	30	500	2020-05-04	70	350	2020-05-06	close	2011041
12	34871009	Shampoo	Complementary	100	5	2020-01-02	30	500	2020-05-05	70	350	2020-05-07	close	2011042
13	87540094	Toilet Brush	Cleaner	90	30	2020-01-02	50	2700	2020-06-03	40	1200	2020-06-05	close	2011043
14	87382933	Detergent	Laundry	100	40	2020-01-03	70	4000	2020-06-05	30	1200	2020-06-07	close	2011044
15	33882970	Dusting Cloth	Cleaning	30	20	2020-01-04	10	600	2020-06-06	10	200	2020-06-07	close	2011045
16	46800229	Soap	Complementary	100	40	2020-01-05	50	4000	2020-06-03	50	2000	2020-06-04	close	2011046
17	577883300	Trays	Daily use	10	20	2020-01-06	6	200	2020-06-03	4	80	2020-06-04	close	2011047
18	12500655	Toilet Paper	Sanitary	100	30	2020-01-01	30	3000	2020-06-04	70	2100	2020-06-05	close	2011048