

AHMED M. S. ALBREEM

Std.id : 210041258

COURSE.NO : CSE4308

DATE OF SOLUTION : 01-09-2023

ASSIGNMENT 3 OF DBMS LAB

FIRST TASK : “creating tables sql statement “

Code :

```
“CREATE TABLE DOCTOR(  
NAME VARCHAR2(20),  
SPECIALIZATION CHAR(2),  
FEE NUMBER,  
PRIMARY KEY (NAME,SPECIALIZATION)  
);  
  
CREATE TABLE PATIENT(  
PATIENT_NO CHAR(5) PRIMARY KEY ,  
NAME VARCHAR(20) NOT NULL,  
ADDERSS VARCHAR2(10)  
);  
  
CREATE TABLE APPOINTMENT(  
PATIENT_NO CHAR(5),  
NAME VARCHAR(20),  
SPECIALIZATION CHAR(2),  
CONSTRAINT PK_APP_PA PRIMARY KEY (PATIENT_NO,NAME,SPECIALIZATION)  
);”
```

The screenshot :

```

1 CREATE TABLE DOCTOR (
2     NAME VARCHAR2(20),
3     SPECIALIZATION CHAR(2),
4     FEE NUMBER,
5     PRIMARY KEY (NAME,SPECIALIZATION)
6 );
7
8 CREATE TABLE PATIENT (
9     PATIENT_NO CHAR(5) PRIMARY KEY,
10    NAME VARCHAR(20) NOT NULL,
11    ADDRESS VARCHAR2(10)
12 );
13
14 CREATE TABLE APPOINTMENT (
15     PATIENT_NO CHAR(5),
16     NAME VARCHAR(20),
17     SPECIALIZATION CHAR(2),
18     CONSTRAINT PK_APP_PA PRIMARY KEY (PATIENT_NO,NAME,SPECIALIZATION)
19 );
20
21 ALTER TABLE APPOINTMENT ADD APPOINTMENT_DATE DATE;
22 ALTER TABLE APPOINTMENT DROP CONSTRAINT PK_APP_PA;
23 ALTER TABLE APPOINTMENT ADD CONSTRAINT NEW_PK_APP_PA PRIMARY KEY (PATIENT_NO,NAME,SPECIALIZATION,APPOINTMENT_DATE);
24
25 ALTER TABLE APPOINTMENT DROP CONSTRAINT PK_APP_PA;
26 ALTER TABLE APPOINTMENT ADD CONSTRAINT NEW_PK_APP_PA PRIMARY KEY (PATIENT_NO,NAME,SPECIALIZATION,APPOINTMENT_DATE);
27
28 ALTER TABLE APPOINTMENT DROP CONSTRAINT PK_APP_PA;
29 ALTER TABLE APPOINTMENT ADD CONSTRAINT NEW_PK_APP_PA PRIMARY KEY (PATIENT_NO,NAME,SPECIALIZATION,APPOINTMENT_DATE);
30 ALTER TABLE APPOINTMENT RENAME COLUMN PATIENT_NO TO P_NO;
31 ALTER TABLE APPOINTMENT RENAME COLUMN NAME TO D_NAME;
32 ALTER TABLE APPOINTMENT RENAME TO APPOINTMENT_INFO;
33
34 /* (**) */
35
36 ALTER TABLE APPOINTMENT_INFO ADD CONSTRAINT FK_APPOINTMENT_D
37 REFERENCES DOCTOR (NAME,SPECIALIZATION) ON DELETE CASCADE;
38
39 ALTER TABLE APPOINTMENT_INFO ADD CONSTRAINT FK_APPOINTMENT_P
40 REFERENCES PATIENT (PATIENT_NO) ON DELETE CASCADE;
41
42 /* (**) */
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

Command Prompt - SQLPLUS

```

SQL> desc PATIENT;
Name Null? Type
-----
PATIENT_NO NOT NULL CHAR(5)
NAME NOT NULL VARCHAR2(20)
ADDRESS VARCHAR2(10)

SQL> desc DOCTOR;
Name Null? Type
-----
NAME NOT NULL VARCHAR2(20)
SPECIALIZATION NOT NULL CHAR(2)
FEE NUMBER

SQL> desc APPOINTMENT;
Name Null? Type
-----
PATIENT_NO NOT NULL CHAR(5)
NAME NOT NULL VARCHAR2(20)
SPECIALIZATION NOT NULL CHAR(2)

```

Second Task : “some operation on the table edition and adding some keys ”

Screenshot in this part : “ i will provide the sql file with doce because its will depend on the creation of the table “

Code 2.a : “ALTER TABLE APPOINTMENT ADD APPOINTMENT_DATE DATE;”

Explain 2.a : “i want to Create new column that include date of APPOINTMENT his input type is DATE ”

Code 2.b : “ALTER TABLE APPOINTMENT DROP CONSTRAINT PK_APP_PA;

ALTER TABLE APPOINTMENT ADD CONSTRAINT NEW_PK_APP_PA PRIMARY KEY (PATIENT_NO,NAME,SPECIALIZATION,APPOINTMENT_DATE);”

Explain 2.b : “i want to drop first thing after that create new one with new values 4 primary keys”

Code 2.c : “ ALTER TABLE APPOINTMENT RENAME COLUMN PATIENT_NO TO P_NO;
ALTER TABLE APPOINTMENT RENAME COLUMN NAME TO D_NAME;”

Explain 2.c : “i want to change the column name from PATIENT_NO to P_NO and also another one from NAME to D_NAME”

Code 2.d : “ ALTER TABLE APPOINTMENT RENAME TO APPOINTMENT_INFO ;”

Explain 2.d : “here i want just to rename the all table name from it name to another name without mentioning the column as <keyword>”

Code 2.e :”ALTER TABLE APPOINTMENT_INFO ADD CONSTRAINT
FK_APPOINTMENT_DOCTOR FOREIGN KEY(D_NAME,SPECIALIZATION)
REFERENCES DOCTOR(NAME,SPECIALIZATION) ON DELETE CASCADE;
ALTER TABLE APPOINTMENT_INFO ADD CONSTRAINT
FK_APPOINTMENT_PATIENT FOREIGN KEY(P_NO)
REFERENCES PATIENT(PATIENT_NO) ON DELETE CASCADE;”

Explain 2.e : ” Here is just editing the relation part of the table between it with another tables to have the connection between the two tables first : FK_APPOINTMENT_DOCTOR is the connect between table of DOCTOR and table APPOINTMENT_INFO ,Second :
FK_APPOINTMENT_PATIENT is the connect between two table which it is
APPOINTMENT_INFO and PATIENT ”

The third part of the assignment : “insert data between the tables to make sure its will work as what planned in the databases ”

Code 3 : “

```
INSERT INTO DOCTOR(NAME,SPECIALIZATION,FEE) VALUES('MR. X','CS',2000);  
INSERT INTO DOCTOR(NAME,SPECIALIZATION,FEE) VALUES('MR. Y','GS',1500);  
INSERT INTO DOCTOR(NAME,SPECIALIZATION,FEE) VALUES('MR. Z','SS',1000);  
INSERT INTO PATIENT(PATIENT_NO,NAME,ADDRSS) VALUES('P-101','A','DHK');  
INSERT INTO PATIENT(PATIENT_NO,NAME,ADDRSS) VALUES('P-102','B','KHL');  
INSERT INTO PATIENT(PATIENT_NO,NAME,ADDRSS) VALUES('P-103','C','SQD');  
INSERT INTO  
APPOINTMENT_INFO(PATIENT_NO,NAME,SPECIALIZATION,APPOINTMENT_DATE)  
VALUES('P-101','MR. X','CS','01-SEP-23');  
INSERT INTO  
APPOINTMENT_INFO(PATIENT_NO,NAME,SPECIALIZATION,APPOINTMENT_DATE)  
VALUES('P-102','MR. Y','GS','02-SEP-23');  
INSERT INTO  
APPOINTMENT_INFO(PATIENT_NO,NAME,SPECIALIZATION,APPOINTMENT_DATE)  
VALUES('P-103','MR. Z','SS','03-SEP-24');  
”
```

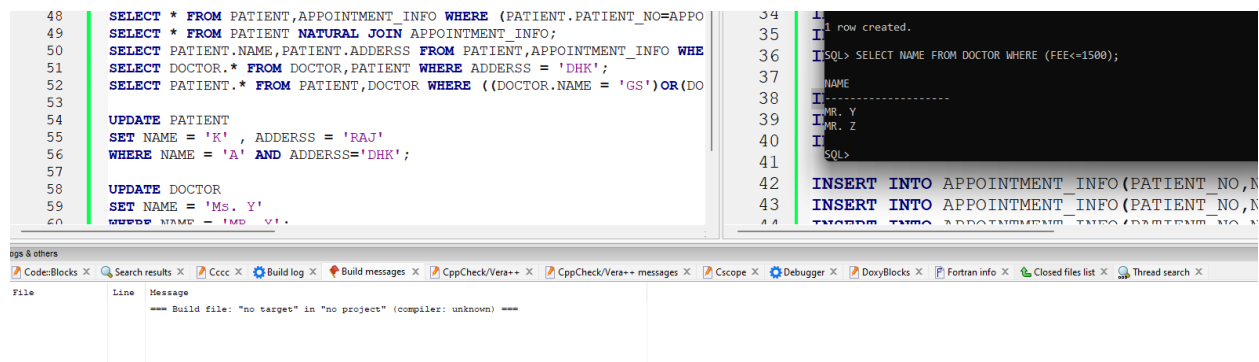
Explain 3 : “here i alter the table by new column as **APPOINTMENT_DATE** just that what i want mention here and all of this just entering information inside the group of tables ”

Fourth Task of the assignment : “some operation to show some data to check the relation between them ”

Explaining : “in this part there is no explaining part just the screenshot of the code because the explaining same as the request of the code ”

Code 4.a : “SELECT NAME FROM DOCTOR WHERE FEE<=1500;”

Screenshot 4.a :



The screenshot shows a code editor with two panels. The left panel contains SQL code for updating and selecting data from the PATIENT and DOCTOR tables. The right panel shows the execution results of a query.

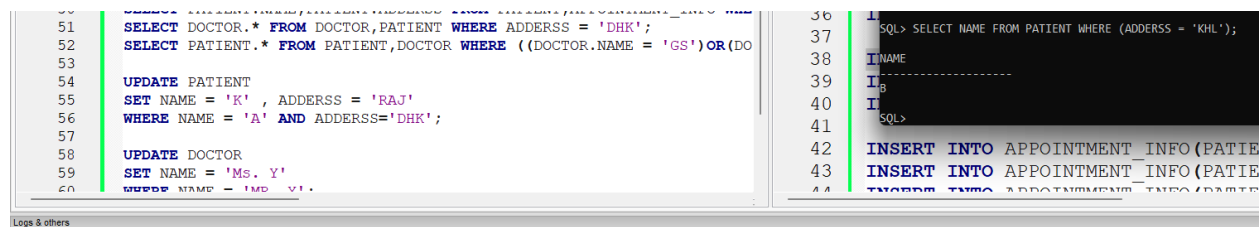
```
48 SELECT * FROM PATIENT,APPOINTMENT_INFO WHERE (PATIENT.PATIENT_NO=APPO
49 SELECT * FROM PATIENT NATURAL JOIN APPOINTMENT_INFO;
50 SELECT PATIENT.NAME,PATIENT.ADDRESS FROM PATIENT,APPOINTMENT_INFO WHE
51 SELECT DOCTOR.* FROM DOCTOR,PATIENT WHERE ADDRESS = 'DHK';
52 SELECT PATIENT.* FROM PATIENT,DOCTOR WHERE ((DOCTOR.NAME = 'GS')OR(DO
53
54 UPDATE PATIENT
55 SET NAME = 'K' , ADDRESS = 'RAJ'
56 WHERE NAME = 'A' AND ADDRESS='DHK';
57
58 UPDATE DOCTOR
59 SET NAME = 'Ms. Y'
60 UPDATE NAME = 'MD V'.
```

```
34 1 row created.
35
36 SQL> SELECT NAME FROM DOCTOR WHERE (FEE<=1500);
37
38 NAME
39 -----
40 MR. Y
41 MR. Z
42
43 SQL>
42 INSERT INTO APPOINTMENT_INFO (PATIENT_NO,N
43 INSERT INTO APPOINTMENT_INFO (PATIENT_NO,N
44 INSERT INTO APPOINTMENT_INFO (PATIENT_NO,N
```

The bottom panel shows a message: "Build file: "no target" in "no project" (compiler: unknown) =="

Code 4.b : “SELECT NAME FROM PATIENT WHERE (ADDRESS = 'KHL');”

Screenshot 4.b :



The screenshot shows a code editor with two panels. The left panel contains SQL code for updating and selecting data from the PATIENT and DOCTOR tables. The right panel shows the execution results of a query.

```
51 SELECT DOCTOR.* FROM DOCTOR,PATIENT WHERE ADDRESS = 'DHK';
52 SELECT PATIENT.* FROM PATIENT,DOCTOR WHERE ((DOCTOR.NAME = 'GS')OR(DO
53
54 UPDATE PATIENT
55 SET NAME = 'K' , ADDRESS = 'RAJ'
56 WHERE NAME = 'A' AND ADDRESS='DHK';
57
58 UPDATE DOCTOR
59 SET NAME = 'Ms. Y'
60 UPDATE NAME = 'MD V'.
```

```
36 SQL> SELECT NAME FROM PATIENT WHERE (ADDRESS = 'KHL');
37
38 NAME
39 -----
40
41 SQL>
42 INSERT INTO APPOINTMENT_INFO (PATIE
43 INSERT INTO APPOINTMENT_INFO (PATIE
44 INSERT INTO APPOINTMENT_INFO (PATIE
```

Code 4.c : “SELECT * FROM PATIENT,APPOINTMENT_INFO WHERE (PATIENT.PATIENT_NO=APPOINTMENT_INFO.PATIENT_NO);”

Screenshot 4.c :

Winter 2022-23

Data Definition and Data Manipulation

(e) Find all the Patient's names and their address who have an appointment with a doctor of 'GS' specialization or a doctor whose fee is greater than 1500.

(f) Find all the Doctor-related information who have an appointment with a doctor of 'GS' specialization or a doctor whose fee is greater than 1500.

(g) Find all Patient-related information who has an appointment with a doctor of 'GS' specialization or a doctor whose fee is greater than 1500.

5. Write following DML statements:

(a) Update the NAME and ADDRESS of a tuple from 'A' and 'DHK' to 'K' and 'RAJ' accordingly.

(b) Update the NAME of table DOCTOR from 'MR. Y' as 'Ms. Y'.

```
SQL> SELECT * FROM PATIENT,APPOINTMENT_INFO WHERE (PATIENT.PATIENT_NO=APPOINTMENT_INFO.PATIENT_NO);
```

PATIENT NAME	ADDRESS	PATIENT NAME	SP	APPOINTMENT DATE
P-101 A	DHK	P-101 MR. X	CS	01-SEP-23
P-102 B	KHL	P-102 MR. Y	GS	02-SEP-23
P-103 C	SQD	P-103 MR. Z	SS	03-SEP-24

Code 4.d : “SELECT * FROM PATIENT NATURAL JOIN APPOINTMENT_INFO;”

Small note : “i have a problem with showing the output here i dont know why but i think its the data type and data and Data Integrity and Column Names ”

Screenshot :

(a) Update the NAME and ADDRESS of a tuple from 'A' and 'DHK' to 'K' and 'RAJ' accordingly.

(b) Update the NAME of table DOCTOR from 'MR. Y' as 'Ms. Y'.

Code 4.d : “SELECT * FROM PATIENT NATURAL JOIN APPOINTMENT_INFO;”

Small note : “i have a problem with showing the output here i dont know why but i think its the data type and data and Data Integrity and Column Names ”

Screenshot :

```
SQL> SELECT * FROM PATIENT NATURAL JOIN APPOINTMENT_INFO;
```

no rows selected

```
SQL>
```

Code 4.e : “SELECT PATIENT.NAME,PATIENT.ADDRESS FROM PATIENT,APPOINTMENT_INFO WHERE APPOINTMENT_INFO.APPOINTMENT_DATE='01-SEP-23';”

Small thing : “here he must show one but showing three the data inserted is different data and i select to show one selected data same as today but it’s not working ”

Screenshot :

```
FREE.sql - CodeBlocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Plugins DovyBlocks Settings Help
SQL SELECT NAME FROM PATIENT WHERE (ADDRESS = 'KHL');
NAME
-----
SQL SELECT * FROM PATIENT,APPOINTMENT_INFO WHERE (PATIENT.PATIENT_NO=APPOINTMENT_INFO.PATIENT_NO);
PATIE NAME      ADDRESS      PATIE NAME      SP APPOINTME
-----
DHK             P-101 MR. X      CS 01-SEP-23
KHL             P-102 MR. Y      GS 02-SEP-23
SQD             P-103 MR. Z      SS 03-SEP-24
SQL SELECT * FROM PATIENT NATURAL JOIN APPOINTMENT_INFO;
no rows selected
SQL SELECT PATIENT.NAME,PATIENT.ADDRESS FROM PATIENT,APPOINTMENT_INFO WHERE APPOINTMENT_INFO.APPOINTMENT_DATE='01-SEP-23';
NAME      ADDRESS
-----
DHK
KHL
SQD
SQL>

33
34 INSERT INTO DOCTOR (NAME,SPECIALIZATION,FEE) VALUES ('MR. X','CS',2000);
35 INSERT INTO DOCTOR (NAME,SPECIALIZATION,FEE) VALUES ('MR. Y','GS',1500);
36 INSERT INTO DOCTOR (NAME,SPECIALIZATION,FEE) VALUES ('MR. Z','SS',1000);
37
38 INSERT INTO PATIENT (PATIENT_NO,NAME,ADDRESS) VALUES ('P-101','A','DHK');
39 INSERT INTO PATIENT (PATIENT_NO,NAME,ADDRESS) VALUES ('P-102','B','KHL');
40 INSERT INTO PATIENT (PATIENT_NO,NAME,ADDRESS) VALUES ('P-103','C','SQD');
41
42 INSERT INTO APPOINTMENT_INFO (PATIENT_NO,NAME,SPECIALIZATION,APPOINTMENT_DATE) VALUES (101,'A','CS','01-SEP-23');
43 INSERT INTO APPOINTMENT_INFO (PATIENT_NO,NAME,SPECIALIZATION,APPOINTMENT_DATE) VALUES (102,'B','GS','02-SEP-23');
44 INSERT INTO APPOINTMENT_INFO (PATIENT_NO,NAME,SPECIALIZATION,APPOINTMENT_DATE) VALUES (103,'C','SS','03-SEP-24');
45
46 SELECT NAME FROM DOCTOR WHERE (FEE<=1500);
47 SELECT NAME FROM PATIENT WHERE (ADDRESS = 'KHL');
48 SELECT * FROM PATIENT,APPOINTMENT_INFO WHERE (PATIENT.PATIENT_NO=APPOINTMENT_INFO.PATIENT_NO);
49 SELECT * FROM PATIENT NATURAL JOIN APPOINTMENT_INFO;
50 SELECT PATIENT.NAME,PATIENT.ADDRESS FROM PATIENT,APPOINTMENT_INFO WHERE APPOINTMENT_INFO.APPOINTMENT_DATE='01-SEP-23';
51 SELECT DOCTOR.* FROM DOCTOR,PATIENT WHERE ADDRESS = 'DHK';
52 SELECT PATIENT.* FROM PATIENT,DOCTOR WHERE ((DOCTOR.NAME = 'GS') AND (PATIENT.ADDRESS = 'DHK'));
53
54 UPDATE PATIENT
55 SET NAME = 'K', ADDRESS = 'RAJ';
56 WHERE NAME = 'A' AND ADDRESS='DHK';
57
58 UPDATE DOCTOR
59 SET NAME = 'Ms. Y';
60 WHERE NAME = 'MD Y';
```

Code 4.f : “SELECT DOCTOR.* FROM DOCTOR,PATIENT WHERE ADDRESS = 'DHK';”

Screenshot :

```
FREE.sql - CodeBlocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Plugins DovyBlocks Settings Help
SQL SELECT NAME FROM PATIENT WHERE (ADDRESS = 'KHL');
NAME
-----
SQL SELECT * FROM PATIENT,APPOINTMENT_INFO WHERE (PATIENT.PATIENT_NO=APPOINTMENT_INFO.PATIENT_NO);
PATIE NAME      ADDRESS      PATIE NAME      SP APPOINTME
-----
DHK             P-101 MR. X      CS 01-SEP-23
KHL             P-102 MR. Y      GS 02-SEP-23
SQD             P-103 MR. Z      SS 03-SEP-24
SQL SELECT * FROM PATIENT NATURAL JOIN APPOINTMENT_INFO;
no rows selected
SQL SELECT PATIENT.NAME,PATIENT.ADDRESS FROM PATIENT,APPOINTMENT_INFO WHERE APPOINTMENT_INFO.APPOINTMENT_DATE='01-SEP-23';
NAME      ADDRESS
-----
DHK
KHL
SQD
SQL>

33
34 INSERT INTO DOCTOR (NAME,SPECIALIZATION,FEE) VALUES ('MR. X','CS',2000);
35 INSERT INTO DOCTOR (NAME,SPECIALIZATION,FEE) VALUES ('MR. Y','GS',1500);
36 INSERT INTO DOCTOR (NAME,SPECIALIZATION,FEE) VALUES ('MR. Z','SS',1000);
37
38 INSERT INTO PATIENT (PATIENT_NO,NAME,ADDRESS) VALUES ('P-101','A','DHK');
39 INSERT INTO PATIENT (PATIENT_NO,NAME,ADDRESS) VALUES ('P-102','B','KHL');
40 INSERT INTO PATIENT (PATIENT_NO,NAME,ADDRESS) VALUES ('P-103','C','SQD');
41
42 INSERT INTO APPOINTMENT_INFO (PATIENT_NO,NAME,SPECIALIZATION,APPOINTMENT_DATE) VALUES (101,'A','CS','01-SEP-23');
43 INSERT INTO APPOINTMENT_INFO (PATIENT_NO,NAME,SPECIALIZATION,APPOINTMENT_DATE) VALUES (102,'B','GS','02-SEP-23');
44 INSERT INTO APPOINTMENT_INFO (PATIENT_NO,NAME,SPECIALIZATION,APPOINTMENT_DATE) VALUES (103,'C','SS','03-SEP-24');
45
46 SELECT NAME FROM DOCTOR WHERE (FEE<=1500);
47 SELECT NAME FROM PATIENT WHERE (ADDRESS = 'KHL');
48 SELECT * FROM PATIENT,APPOINTMENT_INFO WHERE (PATIENT.PATIENT_NO=APPOINTMENT_INFO.PATIENT_NO);
49 SELECT * FROM PATIENT NATURAL JOIN APPOINTMENT_INFO;
50 SELECT PATIENT.NAME,PATIENT.ADDRESS FROM PATIENT,APPOINTMENT_INFO WHERE APPOINTMENT_INFO.APPOINTMENT_DATE='01-SEP-23';
51 SELECT DOCTOR.* FROM DOCTOR,PATIENT WHERE ADDRESS = 'DHK';
52 SELECT PATIENT.* FROM PATIENT,DOCTOR WHERE ((DOCTOR.NAME = 'GS') AND (PATIENT.ADDRESS = 'DHK'));
53
54 UPDATE PATIENT
55 SET NAME = 'K', ADDRESS = 'RAJ';
56 WHERE NAME = 'A' AND ADDRESS='DHK';
57
58 UPDATE DOCTOR
59 SET NAME = 'Ms. Y';
60 WHERE NAME = 'MD Y';
```

Code 4.g: "SELECT PATIENT.* FROM PATIENT,DOCTOR WHERE ((DOCTOR.NAME = 'GS')OR(DOCTOR.FEE >1500));"

Screenshot :

The screenshot shows a code editor with the following SQL code:

```

33
34 INSERT INTO DOCTOR (NAME,SPECIALIZATION,FEE) VALUES ('MR. X','CS',2000);
35 INSERT INTO DOCTOR (NAME,SPECIALIZATION,FEE) VALUES ('MR. Y','GS',1500);
36 INSERT INTO DOCTOR (NAME,SPECIALIZATION,FEE) VALUES ('MR. Z','SS',1000);
37
38 INSERT INTO PATIENT (PATIENT_NO,NAME,ADDRESS) VALUES ('P-101','A','DHK');
39 INSERT INTO PATIENT (PATIENT_NO,NAME,ADDRESS) VALUES ('P-102','B','KHL');
40 INSERT INTO PATIENT (PATIENT_NO,NAME,ADDRESS) VALUES ('P-103','C','SQD');
41
42 INSERT INTO APPOINTMENT_INFO (PATIENT_NO,NAME,SPECIALIZATION,APPOINTMENT_DATE) VALUES ('P-101','A','DHK','01-SEP-23');
43 INSERT INTO APPOINTMENT_INFO (PATIENT_NO,NAME,SPECIALIZATION,APPOINTMENT_DATE) VALUES ('P-102','B','KHL','02-SEP-23');
44 INSERT INTO APPOINTMENT_INFO (PATIENT_NO,NAME,SPECIALIZATION,APPOINTMENT_DATE) VALUES ('P-103','C','SQD','03-SEP-24');
45
46 SELECT NAME FROM DOCTOR WHERE (FEE<=1500);
47 SELECT NAME FROM PATIENT WHERE (ADDRESS = 'KHL');
48 SELECT * FROM PATIENT,APPOINTMENT_INFO WHERE (PATIENT.PATIENT_NO=APPOINTMENT_INFO.PATIENT_NO);
49 SELECT * FROM PATIENT NATURAL JOIN APPOINTMENT_INFO;
50 SELECT PATIENT.NAME,PATIENT.ADDRESS FROM PATIENT,APPOINTMENT_INFO WHERE (PATIENT.PATIENT_NO=APPOINTMENT_INFO.PATIENT_NO);
51 SELECT DOCTOR.* FROM DOCTOR,PATIENT WHERE ADDRESS = 'DHK';
52 SELECT PATIENT.* FROM PATIENT,DOCTOR WHERE ((DOCTOR.NAME = 'GS')OR(DOCTOR.FEE >1500));
53
54 UPDATE PATIENT
55 SET NAME = 'K', ADDRESS = 'RAJ'
56 WHERE NAME = 'A' AND ADDRESS='DHK';
57
58 UPDATE DOCTOR
59 SET NAME = 'Ms. Y'
60 WHERE NAME = 'MR. Y';

```

The Command Prompt window shows the following output for the query in line 52:

```

SQL> SELECT PATIENT.* FROM PATIENT,DOCTOR WHERE ((DOCTOR.NAME = 'GS')OR(DOCTOR.FEE >1500));
PATIENT NAME ADDRESS
-----
P-101 A DHK
P-102 B KHL
P-103 C SQD

```

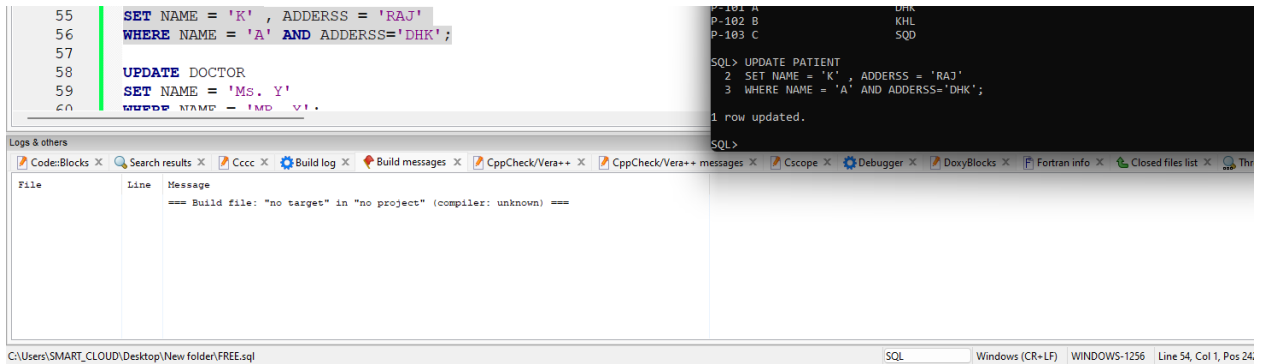

Task five : “data manipulation language ”

The first part is 5.1 : update code : “UPDATE PATIENT

SET NAME = 'K' , ADDERSS = 'RAJ'

WHERE NAME = 'A' AND ADDERSS='DHK';”

Screenshot :



The second part 5.2 : update code : ”

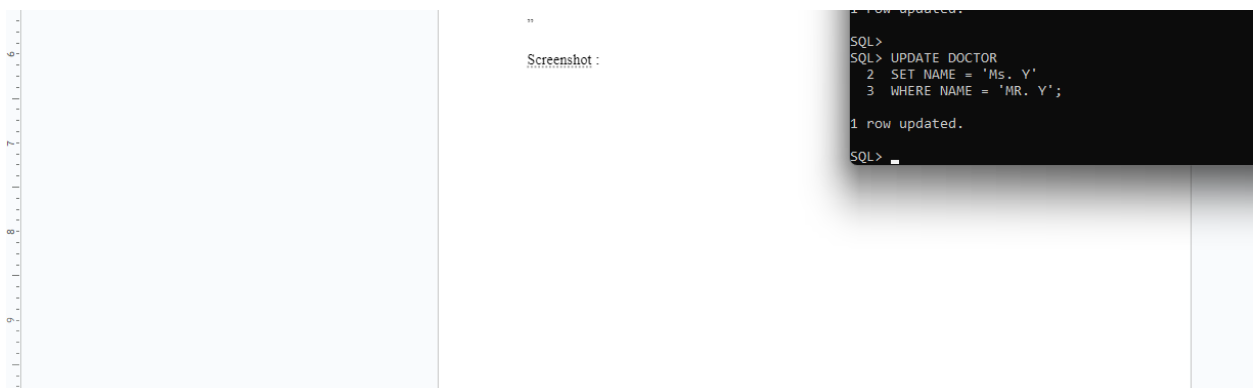
UPDATE DOCTOR

SET NAME = 'Ms. Y'

WHERE NAME = 'MR. Y';

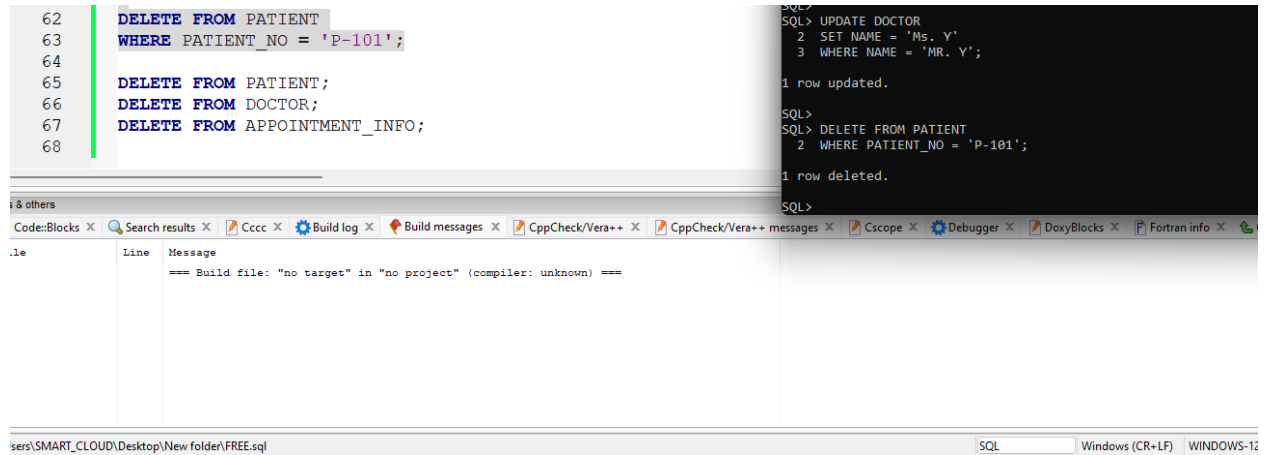
”

Screenshot :



The third part of delete part : code of delete : "DELETE FROM PATIENT
WHERE PATIENT_NO = 'P-101';

”



The screenshot shows a code editor with a dark theme. On the left, a code file named 'FREE.sql' contains the following SQL code:

```
62 DELETE FROM PATIENT
63 WHERE PATIENT_NO = 'P-101';
64
65 DELETE FROM PATIENT;
66 DELETE FROM DOCTOR;
67 DELETE FROM APPOINTMENT_INFO;
68
```

On the right, a terminal window shows the execution of SQL commands:

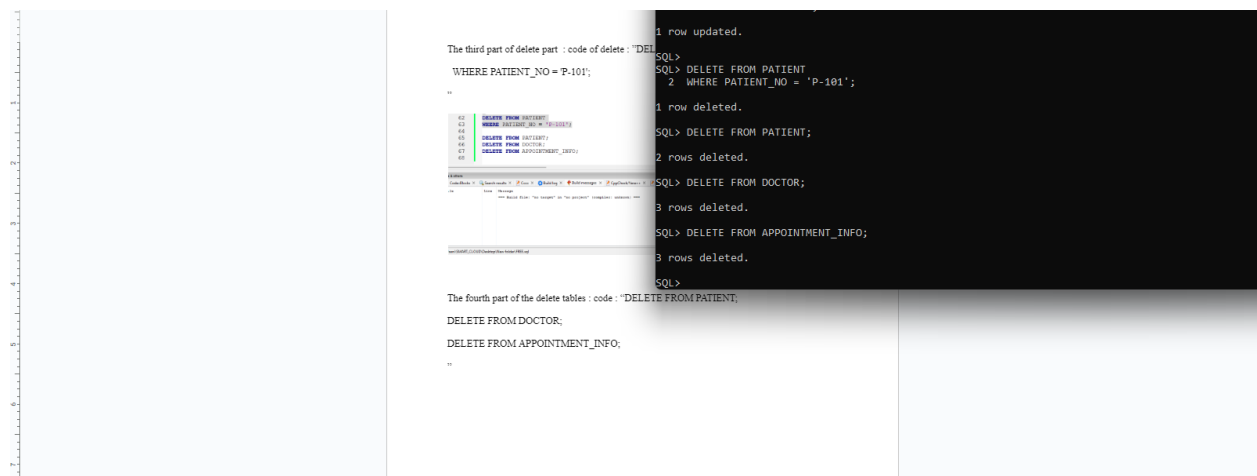
```
SQL> UPDATE DOCTOR
2 SET NAME = 'Ms. Y'
3 WHERE NAME = 'MR. Y';
1 row updated.
SQL>
SQL> DELETE FROM PATIENT
2 WHERE PATIENT_NO = 'P-101';
1 row deleted.
SQL>
```

The bottom of the screenshot shows a status bar with the file path 'sers\SMART_CLOUD\Desktop\New folder\FREE.sql' and tabs for 'SQL', 'Windows (CR+LF)', and 'WINDOWS-12'.

The fourth part of the delete tables : code : “DELETE FROM PATIENT;
DELETE FROM DOCTOR;
DELETE FROM APPOINTMENT_INFO;

”

Screenshot :



The screenshot shows a code editor with a dark theme. On the left, a code file named 'FREE.sql' contains the following SQL code:

```
62 DELETE FROM PATIENT;
63 DELETE FROM DOCTOR;
64 DELETE FROM APPOINTMENT_INFO;
65
66 DELETE FROM PATIENT;
67 DELETE FROM DOCTOR;
68 DELETE FROM APPOINTMENT_INFO;
69
```

On the right, a terminal window shows the execution of SQL commands:

```
1 row updated.
SQL>
SQL> DELETE FROM PATIENT
2 WHERE PATIENT_NO = 'P-101';
1 row deleted.
SQL> DELETE FROM PATIENT;
2 rows deleted.
SQL> DELETE FROM DOCTOR;
3 rows deleted.
SQL> DELETE FROM APPOINTMENT_INFO;
3 rows deleted.
SQL>
```

The bottom of the screenshot shows a status bar with the file path 'sers\SMART_CLOUD\Desktop\New folder\FREE.sql' and tabs for 'SQL', 'Windows (CR+LF)', and 'WINDOWS-12'.

The problem i faced in this week is the software part of oracle 11 gen that was spending so much time but alhamdulillah i finish all the code in time after that i start to build the database ,
thank you for your help