

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 :

Code::Blocks 20.03 window showing SQL code:

```
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,SP)
19 );
20
21 ALTER TABLE APPOINTMENT ADD APPOINTMENT_DATE DATE;
22 ALTER TABLE APPOINTMENT DROP CONSTRAINT PK_APP_PA;
23
24
25
26
27 /* (*~*) */
28 ALTER TABLE APPOINTMENT_INFO ADD CONSTRAINT FK_APPOINTMENT_D
29 REFERENCES DOCTOR(NAME,SPECIALIZATION) ON DELETE CASCADE;
30 ALTER TABLE APPOINTMENT_INFO ADD CONSTRAINT FK_APPOINTMENT_P
31 REFERENCES PATIENT(PATIENT_NO) ON DELETE CASCADE;
32 /* (*~*) */
33
34
35 ORA-02449: unique/primary keys in table referenced by foreign keys
36
37 SQL> desc PATIENT;
38
39 Name Null? Type
40 PATIENT_NO NOT NULL CHAR(5)
41 NAME NOT NULL VARCHAR2(20)
42 ADDRESS VARCHAR2(10)
43
44 SQL> desc DOCTOR;
45
46 Name Null? Type
47 NAME NOT NULL VARCHAR2(20)
48 SPECIALIZATION NOT NULL CHAR(2)
49 FEE NUMBER
50
51 SQL> desc APPOINTMENT;
52
53 Name Null? Type
54 PATIENT_NO NOT NULL CHAR(5)
55 NAME NOT NULL VARCHAR2(20)
56 SPECIALIZATION NOT NULL CHAR(2)
```

Log & others window showing build messages:

```
File Line Message
==== =====
==== =====
==== =====
==== =====
==== =====
==== =====
==== =====
==== =====
==== =====
==== =====
```

Command Prompt window showing table descriptions:

```
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,ADDRESS) VALUES('P-101','A','DHK');
INSERT INTO PATIENT(PATIENT_NO,NAME,ADDRESS) VALUES('P-102','B','KHL');
INSERT INTO PATIENT(PATIENT_NO,NAME,ADDRESS) 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 the Code::Blocks IDE interface. On the left, the code editor displays a script with several SQL statements. On the right, the terminal window shows the execution of the query and its results.

```
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 = 'DHR';
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 WHERE NAME = 'MD VI'.
```

```
34 1 row created.
35
36 SQL> SELECT NAME FROM DOCTOR WHERE (FEE<=1500);
37
38 NAME
39 -----
40 I MR. Y
41 I MR. Z
42 SQL>
43 INSERT INTO APPOINTMENT_INFO(PATIENT_NO,N
44 INSERT INTO APPOINTMENT_INFO(PATIENT_NO,N
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
```

Logs & others

File Line 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 the Code::Blocks IDE interface. On the left, the code editor displays a script with several SQL statements. On the right, the terminal window shows the execution of the query and its results.

```
51
52
53
54
55
56
57
58
59
60
```

```
36 1
37 SQL> SELECT NAME FROM PATIENT WHERE (ADDRESS = 'KHL');
38
39 NAME
40 -----
41 I S
42 SQL>
43 INSERT INTO APPOINTMENT_INFO(PATIE
44 INSERT INTO APPOINTMENT_INFO(PATIE
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
```

Logs & others

Code 4.c : “SELECT \* FROM PATIENT,APPOINTMENT\_INFO WHERE  
(PATIENT.PATIENT\_NO=APPOINTMENT\_INFO.PATIENT\_NO);”

Screenshot 4.c :

4

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

```

Winter 2022-23	Data Definition and Data Manipulation	PATIE NAME	ADDRESS	PATIE NAME	SP_APPOINTME
		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

(e) Find all the Patient's names and their address who has appointment on 01-SEP-23.  
(f) Find all the Doctor-related information who have fee 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:

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

5

```
SQL> SELECT * FROM PATIENT NATURAL JOIN APPOINTMENT_INFO;
no rows selected
SQL>
```

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 :

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 :

The screenshot shows the Code::Blocks IDE interface with two windows open:

- Code::Blocks Window:** Displays the file `FREE.sql` containing SQL code for inserting data into DOCTOR, PATIENT, and APPOINTMENT\_INFO tables, and selecting data from these tables.
- Command Prompt - SQLPLUS Window:** Shows the execution of the SQL code. It includes the following output:

```

SQL> SELECT * FROM PATIENT WHERE ADDRESS = 'KHL';
+-----+
| NAME |
+-----+
| MR. Y|
| MR. Z|
+-----+
SQL> SELECT * FROM PATIENT_APPOINTMENT_INFO WHERE (PATIENT.PATIENT_NO=APPOINTMENT_INFO.PATIENT_NO);
+-----+
| P-101 A          | P-101 B          | P-102 A          | P-102 B          | P-103 C          |
| DMR             | DMR             | KHL             | KHL             | SQD             |
| P-101 MR. X     | P-101 MR. X     | P-102 MR. Y     | P-102 MR. Y     | P-103 MR. Z    |
| CS 01-SEP-23   | CS 01-SEP-23   | GS 02-SEP-23   | GS 02-SEP-23   | SS 03-SEP-24   |
+-----+
SQL> SELECT * FROM PATIENT NATURAL JOIN APPOINTMENT_INFO;
no rows selected
SQL> SELECT * FROM PATIENT_APPOINTMENT_INFO WHERE APPOINTMENT_DATE='01-SEP-23';
+-----+
| NAME |
+-----+
| DRK |
| KHL |
| SQD |
+-----+
SQL> SELECT DOCTOR.* FROM DOCTOR, PATIENT WHERE ADDRESS = 'DHK';
+-----+
| NAME |
+-----+
| DHK |
| KHL |
| SQD |
+-----+
SQL>

```

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

## Screenshot :

The screenshot shows the Code::Blocks IDE interface with two windows open:

- Code::Blocks Window:** Displays the file `FREE.sql` containing SQL code for inserting data into DOCTOR, PATIENT, and APPOINTMENT\_INFO tables, and selecting data from these tables.
- Command Prompt - SQLPLUS Window:** Shows the execution of the SQL code. It includes the following output:

```

SQL> SELECT * FROM PATIENT WHERE ADDRESS = 'KHL';
+-----+
| NAME |
+-----+
| MR. Y|
| MR. Z|
+-----+
SQL> SELECT * FROM PATIENT_APPOINTMENT_INFO WHERE (PATIENT.PATIENT_NO=APPOINTMENT_INFO.PATIENT_NO);
+-----+
| P-101 A          | P-101 B          | P-102 A          | P-102 B          | P-103 C          |
| DMR             | DMR             | KHL             | KHL             | SQD             |
| P-101 MR. X     | P-101 MR. X     | P-102 MR. Y     | P-102 MR. Y     | P-103 MR. Z    |
| CS 01-SEP-23   | CS 01-SEP-23   | GS 02-SEP-23   | GS 02-SEP-23   | SS 03-SEP-24   |
+-----+
SQL> SELECT * FROM PATIENT NATURAL JOIN APPOINTMENT_INFO;
no rows selected
SQL> SELECT * FROM PATIENT_APPOINTMENT_INFO WHERE APPOINTMENT_DATE='01-SEP-23';
+-----+
| NAME |
+-----+
| DRK |
| KHL |
| SQD |
+-----+
SQL> SELECT DOCTOR.* FROM DOCTOR, PATIENT WHERE ADDRESS = 'DHK';
+-----+
| NAME |
+-----+
| DRK |
| KHL |
| SQD |
+-----+
SQL> SELECT DOCTOR.* FROM DOCTOR, PATIENT WHERE ADDRESS = 'DHK';
+-----+
| NAME | SP | FEE |
+-----+
| DRK |   | 2000 |
| KHL |   | 1500 |
| SQD |   | 1000 |
+-----+
SQL>

```

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

Screenshot :

The screenshot shows the CodeBlocks IDE interface with two windows open:

- FREE.sql - CodeBlocks 20.03**: This window contains the SQL code for the query. It includes several INSERT statements for DOCTOR and PATIENT tables, and a complex SELECT statement involving PATIENT, APPOINTMENT\_INFO, and DOCTOR tables.
- Command Prompt - SQLPLUS**: This window displays the results of the executed SQL query. The results are presented in two parts:
  - A header row: **PATIE NAME ADDRESS SP APPOINTME**
  - Data rows:
 

PATIE NAME	ADDRESS	SP	APPOINTME
P-101 A	DHK	P-101	MR. X
P-102 A	KHL	P-102	MR. Y
P-103 C	SQD	P-103	MR. Z
  - A second header row: **NAME ADDRESS**
  - Data rows:
 

NAME	ADDRESS
MR. X	CS
MR. Y	GS
MR. Z	SS

Below the command prompt window, the status bar shows the path **C:\Users\SMART\_CLOUD\Desktop\New folder\FREE.sql**, the tab **SQL**, and the status **Windows (CR+LF) | WINDOWS-1256 | Line 53, Col 1, Pos 2427 | Insert | Read/Write default**.

Task five : “data manipulation language ”

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

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

WHERE NAME = 'A' AND ADDRESS='DHK';"

## Screenshot :

The screenshot shows the Code::Blocks IDE interface. On the left, there is a code editor with two tabs: 'Logs & others' and 'SQL & others'. The 'SQL & others' tab contains the following SQL code:

```
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 = 'IMP' .
```

The output window on the right shows the results of the SQL queries:

```
P-101 A          DRK
P-102 B          KHL
P-103 C          SQD

SQL> UPDATE PATIENT
  2  SET NAME = 'K' , ADDRESS = 'RAJ'
  3 WHERE NAME = 'A' AND ADDRESS='DHK';

1 row updated.
```

Below the code editor, the status bar displays the file path: C:\Users\SMART\_CLOUD\Desktop\New folder\FREE.sql. The bottom status bar also shows the current window configuration: Windows (CR+LF), WINDOWS-1256, Line 54, Col 1, Pos 24.

The second part 5.2 : update code : "

## UPDATE DOCTOR

SET NAME = 'Ms. Y'

WHERE NAME = 'MR. Y',

22

## Screenshot :

```
""
Screenshot :
SQL> UPDATE DOCTOR
2 SET NAME = 'Ms. Y'
3 WHERE NAME = 'MR. Y';

1 row updated.

SQL> -
```

The third part of delete part : code of delete : "DELETE FROM PATIENT

WHERE PATIENT\_NO = 'P-101';

"

Code::Blocks IDE screenshot showing SQL queries and their results. The left pane shows the code in a file named FREE.sql:

```
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
```

The right pane shows the SQL prompt and its responses:

```
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.
```

Below the code editor, the status bar shows the file path: users\SMART\_CLOUD\Desktop\New folder\FREE.sql, and the tabs 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 :

Oracle SQL\*Plus screenshot showing the execution of the fourth part of the delete tables code:

```
The third part of delete part : code of delete : "DELETE FROM PATIENT WHERE PATIENT_NO = 'P-101';"
"
SQL> DELETE FROM PATIENT
  2  WHERE PATIENT_NO = 'P-101';
1 row updated.

SQL>
SQL> DELETE FROM PATIENT;
2 rows deleted.

SQL> DELETE FROM DOCTOR;
3 rows deleted.

SQL> DELETE FROM APPOINTMENT_INFO;
3 rows deleted.

SQL>
```

The output shows the deletion of one row from the PATIENT table, two rows from the DOCTOR table, and three rows from the APPOINTMENT\_INFO table.

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