

## דו"ח בסיסי נתונים עבור בית חולים "חדר ניתוח"

Eliaou Madar 1954753

Haim Hai 345516224

### Project description:

Our hospital has several operating rooms equipped for a variety of complex surgical procedures, managed by a dedicated team of doctors and nurses ensuring their smooth operation and patient safety. Each room can accommodate multiple operations, planned to minimize patient waiting times. A doctor may perform multiple operations, while a patient may undergo multiple operations during their hospital stay. Nurses, playing a key role, can assist in several operations and are also responsible for maintaining equipment. Additionally, each piece of equipment can be used for multiple operations, thereby optimizing hospital resources.

### Description of entities:

#### 1. Patient (חולה):

- Patient\_ID (PK) - Patient's identification number
- Patient\_Name - The patient name
- Sexe - The sexe of the patient
- Illness - Brief description of the subject of the operation.

#### 2. Operation (ניתוח):

- Operation\_ID (PK) - Operation's identification number
- Operation\_Date - The date of the operation
- Duration\_Operation - The time that takes the operation

### **3. Operating Room (חדר ניתוח):**

- Room\_ID (PK) - Room's identification number
- Availability - Indicates if the room is available.
- Max\_number\_people - indicates the maximum number of people that the room can accommodate

### **4. Equipment (ציוד):**

- Equipment\_ID (PK) - Equipment's identification number
- Equipment\_Name - The equipment name
- Equipment\_Status - Indicates whether the equipment is available.
- Equipment\_Purchase\_Date - Date of purchase of the equipment

### **5. Nurse (אחות):**

- Nurse\_ID (PK) - Nurse's identification number
- Nurse\_Name - The nurse name
- Telephone\_number - The telephone number of the nurse

### **6. Doctor (רופא):**

- Doctor\_ID (PK) - Doctor's identification number

- Doctor\_Name - The doctor name
- Specialty - The doctor's specialty

## **7. Operate\_by:**

- Doctor\_ID (FK) - Doctor's identification number
- Operation\_ID (FK) - Operation's identification number

## **8. Assist\_by:**

- Nurse\_ID (FK) - Nurse's identification number
- Operation\_ID (FK) - Operation's identification number

## **Description of the relationships between the entities:**

A doctor can perform several operations. (M: N)  
An operation is carried out by one or more doctors.

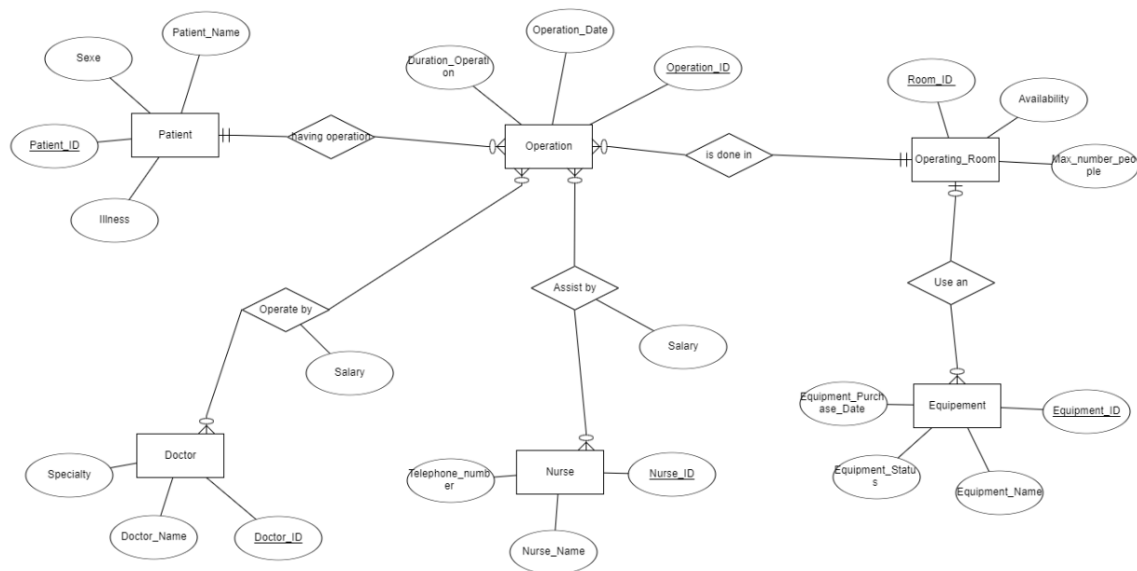
An operation takes place in a single room. (M: 1)  
A room can accommodate several operations.

A patient may undergo several operations. (1: N)  
An operation concerns a single patient.

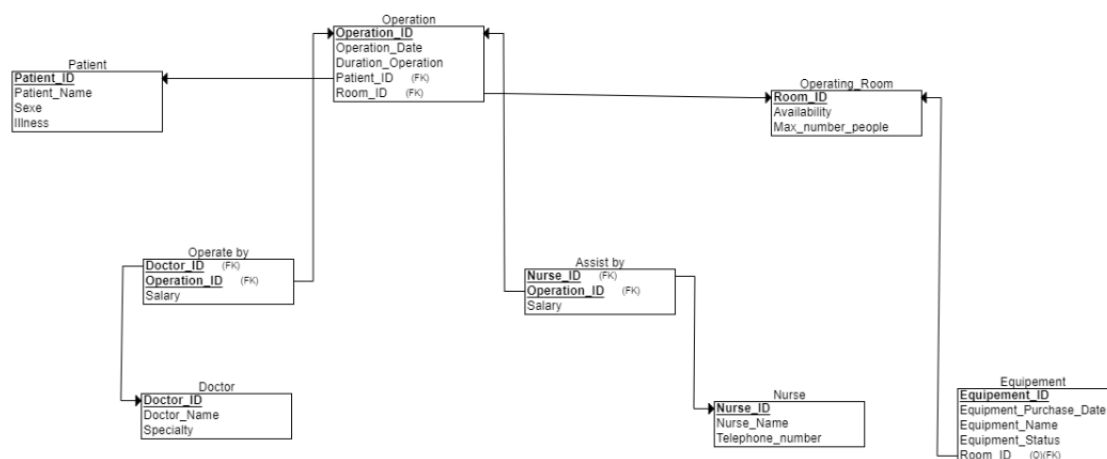
One piece of equipment can be used by a maximum of one operations room. (1: N)  
An operating room may require several pieces of equipment.

A nurse can attend several operations. (M: N)  
 An operation can be assisted by several nurses.

## ERD diagram:



## DSD diagram:



All tables are at 3NF level and there is no need for further normalization. We will prove it:

All tables are in 1NF because all fields are atomic.

All tables respect 2NF, because each table has a unique key. Therefore, no column depends on part of the key, but on the entire key.

All tables respect 3NF: there is no dependency between the different fields, the only dependency being that of the primary key.

Creating the tables:

Creating the **Patient** table:

```

-- Create table
create table PATIENT
(
  patient_id    INTEGER not null,
  sexe          VARCHAR2(30) not null,
  patient_name  VARCHAR2(30) not null,
  illness       VARCHAR2(100) not null
)
tablespace SYSTEM
pctfree 10
pctused 40
initrans 1
maxtrans 255
storage
(
  initial 64K
  next 1M
  minextents 1
  maxextents unlimited
);
-- Create/Recreate primary, unique and foreign key constraints
alter table PATIENT
add primary key (PATIENT_ID)
using index
tablespace SYSTEM
pctfree 10
initrans 2
maxtrans 255
storage
(

```

General

Columns

Keys

Checks

Indexes

Privileges

Triggers

Name	Type	Columns	Enabled	Referencing table	Referencing columns	On Delete	Deferrable	Deferred	Validated	Last change
▶ SYS_C008318	Primary	PATIENT_ID	<input checked="" type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	27/05/2024 15:14:36

General

COL

Columns

Keys

Checks

Indexes

Privileges

Triggers

Type owner

Name

	Name	Virtual	Type	Nullable	Default/Expr.	Generated	On Null	Invisible	Storage	Comments
▶	PATIENT_ID	<input type="checkbox"/>	INTEGER	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>		
	SEXE	<input type="checkbox"/>	VARCHAR2(30)	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>		
	PATIENT_NAME	<input type="checkbox"/>	VARCHAR2(30)	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>		
	ILLNESS	<input type="checkbox"/>	VARCHAR2(100)	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>		

Creating the **Operation** table:

General Columns Keys Checks Indexes Privileges Triggers										
Type owner	Name									
Name	Virtual	Type	Nullable	Default/Expr.	Generated	On Null	Invisible	Storage	Comments	
▶ OPERATION_DATE	<input type="checkbox"/>	DATE	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>			
DURATION_OPERATION	<input type="checkbox"/>	INTEGER	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>			
OPERATION_ID	<input type="checkbox"/>	INTEGER	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>			
PATIENT_ID	<input type="checkbox"/>	INTEGER	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>			
ROOM_ID	<input type="checkbox"/>	INTEGER	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>			

General Columns Keys Checks Indexes Privileges Triggers										
Name	Type	Columns	Enabled	Referencing table	Referencing columns	On Delete	Deferrable	Deferred	Validated	Last change
▶ SYS_C008399	Primary	OPERATION_ID	<input checked="" type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	27/05/2024 20:47:52
SYS_C008400	Foreign	PATIENT_ID	<input checked="" type="checkbox"/>	PATIENT	PATIENT_ID	No action	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	27/05/2024 20:47:52
SYS_C008401	Foreign	ROOM_ID	<input checked="" type="checkbox"/>	OPERATING_ROOM	ROOM_ID	No action	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	27/05/2024 20:47:52

```
-- Create table
create table OPERATION
(
    operation_date      DATE not null,
    duration_operation  INTEGER not null,
    operation_id        INTEGER not null,
    patient_id          INTEGER not null,
    room_id             INTEGER not null
)
tablespace SYSTEM
pctfree 10
pctused 40
initrans 1
maxtrans 255
storage
(
    initial 64K
    next 1M
    minextents 1
    maxextents unlimited
);

-- Create/Recreate primary, unique and foreign key constraints
alter table OPERATION
add primary key (OPERATION_ID)
using index
tablespace SYSTEM
pctfree 10
initrans 2
maxtrans 255
storage
```

Creating the **Operating Room** table:

General

Columns

Keys

Checks

Indexes

Privileges

Triggers

Name	Type	Columns	Enabled	Referencing table	Referencing columns	On Delete	Deferrable	Deferred	Validated	Last change
▶ SYS_C008379	Primary	ROOM_ID	✓				<input type="checkbox"/>	<input type="checkbox"/>	✓	27/05/2024 20:47:52

General

Columns

Keys

Checks

Indexes

Privileges

Triggers

Type owner

Name

Name	Virtual	Type	Nullable	Default/Expr.	Generated	On Null	Invisible	Storage	Comments
▶ ROOM_ID	<input type="checkbox"/>	INTEGER	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>		
AVAILABILITY	<input type="checkbox"/>	VARCHAR2(10)	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>		
MAX_NUMBER_PEOPLE	<input type="checkbox"/>	INTEGER	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>		

---

```

-- Create table
create table OPERATING_ROOM
(
    room_id            INTEGER not null,
    availability        VARCHAR2(10) not null,
    max_number_people  INTEGER not null
)
tablespace SYSTEM
pctfree 10
pctused 40
initrans 1
maxtrans 255
storage
(
    initial 64K
    next 1M
    minextents 1
    maxextents unlimited
);
-- Create/Recreate primary, unique and foreign key constraints
alter table OPERATING_ROOM
add primary key (ROOM_ID)
using index
tablespace SYSTEM
pctfree 10
initrans 2
maxtrans 255
storage
(
    initial 64K

```

Creating the **Equipement** table:



General Columns Keys Checks Indexes Privileges Triggers										
Name	Type	Columns	Enabled	Referencing table	Referencing columns	On Delete	Deferrable	Deferred	Validated	Last change
► SYS_C008384	Primary	EQUIPEMENT_ID	✓				<input type="checkbox"/>	<input type="checkbox"/>	✓	27/05/2024 20:47:52
SYS_C008385	Foreign	ROOM_ID	✓	OPERATING_ROOM	ROOM_ID	No action	<input type="checkbox"/>	<input type="checkbox"/>	✓	27/05/2024 20:47:52

General Columns Keys Checks Indexes Privileges Triggers										
Type owner	Name									
Name	Virtual	Type	Nullable	Default/Expr.	Generated	On Null	Invisible	Storage	Comments	
► EQUIPEMENT_ID	<input type="checkbox"/>	INTEGER	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>			
EQUIPEMENT_PURCHASE_DATE	<input type="checkbox"/>	DATE	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>			
EQUIPEMENT_NAME	<input type="checkbox"/>	VARCHAR2(30)	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>			
EQUIPEMENT_STATUS	<input type="checkbox"/>	VARCHAR2(30)	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>			
ROOM_ID	<input type="checkbox"/>	INTEGER	✓			<input type="checkbox"/>	<input type="checkbox"/>			

```
-- Create table
create table EQUIPEMENT
(
    equipment_id          INTEGER not null,
    equipment_purchase_date DATE not null,
    equipment_name         VARCHAR2(30) not null,
    equipment_status       VARCHAR2(30) not null,
    room_id               INTEGER
)
tablespace SYSTEM
pctfree 10
pctused 40
initrans 1
maxtrans 255
storage
(
    initial 64K
    next 1M
    minextents 1
    maxextents unlimited
);
-- Create/Recreate primary, unique and foreign key constraints
alter table EQUIPEMENT
add primary key (EQUIPEMENT_ID)
using index
tablespace SYSTEM
pctfree 10
initrans 2
maxtrans 255
storage
```

Creating the **Nurse** table:

General

Columns

Keys

Checks

Indexes

Privileges

Triggers

Type owner

Name

Name	Virtual	Type	Nullable	Default/Expr.	Generated	On Null	Invisible	Storage	Comments
NURSE_NAME	<input type="checkbox"/>	VARCHAR2(30)	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>		
TELEPHONE_NUMBER	<input type="checkbox"/>	INTEGER	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>		
NURSE_ID	<input type="checkbox"/>	INTEGER	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>		

General

Columns

Keys

Checks

Indexes

Privileges

Triggers

Name	Type	Columns	Enabled	Referencing table	Referencing columns	On Delete	Deferrable	Deferred	Validated	Last change
SYS_C008393	Primary	NURSE_ID	<input checked="" type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	27/05/2024 20:47:52

```

-- Create table
create table NURSE
(
  nurse_name          VARCHAR2(30) not null,
  telephone_number    INTEGER not null,
  nurse_id            INTEGER not null
)
tablespace SYSTEM
pctfree 10
pctused 40
initrans 1
maxtrans 255
storage
(
  initial 64K
  next 1M
  minextents 1
  maxextents unlimited
);
-- Create/Recreate primary, unique and foreign key constraints
alter table NURSE
add primary key (NURSE_ID)
using index
tablespace SYSTEM
pctfree 10
initrans 2
maxtrans 255
storage
(
  initial 64K

```

Creating the **Doctor** table:

General Columns Keys Checks Indexes Privileges Triggers										
Type owner	Name									
Name	Virtual	Type	Nullable	Default/Expr.	Generated	On Null	Invisible	Storage	Comments	
▶ DOCTOR_NAME	<input type="checkbox"/>	VARCHAR2(30)	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>			
SPECIALTY	<input type="checkbox"/>	VARCHAR2(50)	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>			
DOCTOR_ID	<input type="checkbox"/>	INTEGER	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>			

General Columns Keys Checks Indexes Privileges Triggers										
Name	Type	Columns	Enabled	Referencing table	Referencing columns	On Delete	Deferrable	Deferred	Validated	Last change
▶ SYS_C008389	Primary	DOCTOR_ID	<input checked="" type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	27/05/2024 20:47:52

```
-- Create table
create table DOCTOR
(
    doctor_name VARCHAR2(30) not null,
    specialty   VARCHAR2(50) not null,
    doctor_id   INTEGER not null
)
tablespace SYSTEM
pctfree 10
pctused 40
initrans 1
maxtrans 255
storage
(
    initial 64K
    next 1M
    minextents 1
    maxextents unlimited
);
-- Create/Recreate primary, unique and foreign key constraints
alter table DOCTOR
add primary key (DOCTOR_ID)
using index
tablespace SYSTEM
pctfree 10
initrans 2
maxtrans 255
storage
(
    initial 64K
```

Creating the **Assist by** table:

General

Columns

Keys

Checks

Indexes

Privileges

Triggers

Type owner

Name

Name	Virtual	Type	Nullable	Default/Expr.	Generated	On Null	Invisible	Storage	Comments
SALARY	<input type="checkbox"/>	FLOAT	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>		
NURSE_ID	<input type="checkbox"/>	INTEGER	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>		
OPERATION_ID	<input type="checkbox"/>	INTEGER	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>		

General

Columns

Keys

Checks

Indexes

Privileges

Triggers

Name	Type	Columns	Enabled	Referencing table	Referencing columns	On Delete	Deferrable	Deferred	Validated	Last change
SYS_C008411	Primary	NURSE_ID, OPERATION_ID	<input checked="" type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	27/05/2024 20:47:52
SYS_C008412	Foreign	NURSE_ID	<input checked="" type="checkbox"/>	NURSE	NURSE_ID	No action	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	27/05/2024 20:47:52
SYS_C008413	Foreign	OPERATION_ID	<input checked="" type="checkbox"/>	OPERATION	OPERATION_ID	No action	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	27/05/2024 20:47:52

```
-- Create table
create table ASSIST_BY
(
    salary          FLOAT not null,
    nurse_id        INTEGER not null,
    operation_id     INTEGER not null
)
tablespace SYSTEM
pctfree 10
pctused 40
initrans 1
maxtrans 255
storage
(
    initial 64K
    next 1M
    minextents 1
    maxextents unlimited
);

-- Create/Recreate primary, unique and foreign key constraints
alter table ASSIST_BY
add primary key (NURSE_ID, OPERATION_ID)
using index
tablespace SYSTEM
pctfree 10
initrans 2
maxtrans 255
storage
(
    initial 64K
```

## Creating the **Operate by** table:

GeneralColumnsKeysChecksIndexesPrivilegesTriggers

Type ownerName

Name	Virtual	Type	Nullable	Default/Expr.	Generated	On Null	Invisible	Storage	Comments
SALARY	<input type="checkbox"/>	FLOAT	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>		
DOCTOR_ID	<input type="checkbox"/>	INTEGER	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>		
OPERATION_ID	<input type="checkbox"/>	INTEGER	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>		

GeneralColumnsKeysChecksIndexesPrivilegesTriggers

Name	Type	Columns	Enabled	Referencing table	Referencing columns	On Delete	Deferrable	Deferred	Validated	Last change
SYS_C008405	Primary	DOCTOR_ID, OPERATION_ID	<input checked="" type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	27/05/2024 20:47:52
SYS_C008406	Foreign	DOCTOR_ID	<input checked="" type="checkbox"/>	DOCTOR	DOCTOR_ID	No action	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	27/05/2024 20:47:52
SYS_C008407	Foreign	OPERATION_ID	<input checked="" type="checkbox"/>	OPERATION	OPERATION_ID	No action	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	27/05/2024 20:47:52

```
-- Create table
create table OPERATE_BY
(
    salary          FLOAT not null,
    doctor_id       INTEGER not null,
    operation_id    INTEGER not null
)
tablespace SYSTEM
pctfree 10
pctused 40
initrans 1
maxtrans 255
storage
(
    initial 64K
    next 1M
    minextents 1
    maxextents unlimited
);
-- Create/Recreate primary, unique and foreign key constraints
alter table OPERATE_BY
add primary key (DOCTOR_ID, OPERATION_ID)
using index
tablespace SYSTEM
pctfree 10
initrans 2
maxtrans 255
storage
(
    initial 64K
```

## Entering data by GENERATOR DATA.

Entering data into the Patient table:

PATIENT

< Owner Table Number of records  
> C##ELIAOU\_HAIM PATIENT 400..500  
...

Name	Type	Size	Data
PATIENT_ID	NUMBER		Sequence(100, 1, 1200)
PATIENT_NAME	VARCHAR2	30	LastName + ' ' + FirstName
SEXE	VARCHAR2	30	List('Man', 'Woman')
ILLNESS	VARCHAR2	100	List('appendicitis', 'gallstones', 'hernia', 'colon cancer', 'gastric cancer', 'esophageal cancer', 'panc

SQL Output Statistics

```
select * from patient
```

SQL Output Statistics

	PATIENT_ID	PATIENT_NAME	SEXE	ILLNESS
1	428	Place Beth	Man	appendicitis
2	429	Birch Freddie	Woman	craniotomy
3	430	LaSalle Clea	Woman	hip replacement
4	431	Popper Terrence	Woman	adrenalectomy
5	432	Dalley Mindy	Man	cleft lip and palate
6	433	Rosas Tori	Woman	melanoma
7	434	Bragg Kazem	Man	coronary artery disease
8	435	Sizemore Billy	Man	kidney cancer
9	436	Crouch Helen	Man	bladder cancer
10	437	LaMond Mekhi	Woman	lung cancer
11	438	Isaacs Mac	Man	cholecystectomy
12	439	Goldblum Kristin	Woman	biliary atresia
13	440	Berkoff Roy	Man	thyroidectomy
14	441	Meyer David	Woman	kidney cancer
15	442	Ruffalo Mel	Man	thyroid cancer
16	443	Crouse Grace	Man	craniotomy
17	444	Joli Rupert	Woman	hernia
18	445	Kramer Gordon	Man	spinal fusion
19	446	Fehr Matt	Woman	cleft lip and palate
20	447	Liu Merillee	Man	cervical cancer
21	448	Pavmer Mark	Man	bladder cancer

Entering data into the Operating\_Room table:

OPERATING\_ROOM

< Owner Table Number of records

> C##ELIAOU\_HAIM OPERATING\_ROOM 400..500

...

Name	Type	Size	Data
ROOM_ID	NUMBER		Sequence(1200, 1, 2400)
AVAILABILITY	VARCHAR2	10	List('Yes', 'No')
MAX_NUMBER_PEOPLE	NUMBER		Random(0, 20)

SQL

Output

Statistics

select \* from Operating\_Room

	ROOM_ID	AVAILABILITY	MAX_NUMBER_PEOPLE
▶	1	1200 No	1
	2	1201 No	20
	3	1202 Yes	2
	4	1203 Yes	11
	5	1204 No	5
	6	1205 Yes	7
	7	1206 Yes	6
	8	1207 Yes	5
	9	1208 Yes	2
	10	1209 No	16
	11	1210 No	14
	12	1211 No	12
	13	1212 Yes	9
	14	1213 No	15
	15	1214 No	9
	16	1215 No	16
	17	1216 Yes	15
	18	1217 Yes	5
	19	1218 No	17
	20	1219 Yes	4
	21	1220 Yes	0

### Entering data into the Equipement table:

EQUIPEMENT

<

Owner

Table

Number of records

>

C##ELIAOU\_HAIM

EQUIPEMENT

400..500

...

Name	Type	Size	Data
EQUIPEMENT_ID	NUMBER		Sequence(4800, 1, 6000)
EQUIPMENT_PURCHASE_DATE	DATE		List('01/01/2024', '14/01/2024', '28/01/2024', '05/02/2024', '19/02/2024')
EQUIPMENT_NAME	VARCHAR2	30	List('Scalpel', 'Forceps', 'Hemostat', 'Surgical scissors', 'Needle holder', 'Suture')
EQUIPMENT_STATUS	VARCHAR2	30	List('available', 'not available')
ROOM_ID	NUMBER		List(select ROOM_ID from Operating_Room)

SQL

Output

Statistics

```
select * from Equipement
```

	EQUIPEMENT_ID	EQUIPMENT_PURCHASE_DATE	EQUIPMENT_NAME	EQUIPMENT_STATUS	ROOM_ID
1	4800	10/08/2024	Surgical sutures	available	1431
2	4801	07/09/2024	Surgical gloves	not available	1658
3	4802	05/05/2024	Surgical microscope	not available	1307
4	4803	24/02/2024	Doppler	not available	1402
5	4804	21/09/2024	C-arm	not available	1283
6	4805	20/04/2024	Bone saw	not available	1456
7	4806	09/03/2024	Retractor	not available	1582
8	4807	23/03/2024	Orthopedic implants	available	1652
9	4808	30/06/2024	Suction device	not available	1390
10	4809	04/05/2024	Laser scalpel	not available	1468
11	4810	27/07/2024	Foot pedal	not available	1372
12	4811	15/06/2024	Laparoscope	available	1451
13	4812	09/03/2024	Catheter	available	1427
14	4813	28/07/2024	Surgical tray	available	1274
15	4814	22/09/2024	Scalpel	not available	1624
16	4815	06/10/2024	Retractor	not available	1453
17	4816	09/03/2024	Electrocautery unit	not available	1649
18	4817	05/10/2024	Sterilizer	not available	1226
19	4818	19/02/2024	Hemostat	available	1568
20	4819	09/03/2024	Surgical mask	not available	1496
21	4820	28/01/2024	Surgical gown	not available	1569



## Entering data into the Doctor table:

DOCTOR

<

Owner

Table

Number of records

>

C##ELIAOU\_HAIM

DOCTOR

400..500

...

Name	Type	Size	Data
DOCTOR_NAME	VARCHAR2	30	LastName + ' ' + FirstName
SPECIALTY	VARCHAR2	50	List('Cardiothoracic Surgery', 'General Surgery',
DOCTOR_ID	NUMBER		Sequence(2400, 1, 3600)

SQL

Output

Statistics

select \* from Doctor

	DOCTOR_NAME	SPECIALTY	DOCTOR_ID
1	Laurie Jimmy	General Surgery	2400
2	Gill Giancarlo	Cardiothoracic Surgery	2401
3	Hingle Amy	Reconstructive Surgery	2402
4	Margolyes Alan	Reproductive Surgery	2403
5	Rickles Demi	Pediatric Plastic Surgery	2404
6	DiFranco Jesse	Vascular Surgery	2405
7	Newton Regina	Trauma Surgery	2406
8	Silverman Marina	Neurointerventional Surgery	2407
9	Roundtree Debi	Minimally Invasive Surgery	2408
10	Dourif Bette	Joint Replacement Surgery	2409
11	Hauer Anna	Neurosurgery	2410
12	Quinlan Tia	Cardiac Surgery	2411
13	Paxton Natacha	Hernia Surgery	2412
14	Day-Lewis Jimmy	Pediatric Plastic Surgery	2413
15	Zahn Willie	Foot and Ankle Surgery	2414
16	Plummer Debby	Robotic Surgery	2415
17	Downie Alicia	Spine Surgery	2416
18	Paymer Anne	Hepatobiliary Surgery	2417
19	Walken Isaac	Pediatric Transplant Surgery	2418
20	Newton Glenn	Foot and Ankle Surgery	2419
21	Rea Billy	Joint Replacement Surgery	2420
22	Gore Lou	Pediatric Gastrointestinal Surgery	2421

Entering data into the Nurse table:

NURSE

< Owner Table Number of records

> C##ELIAOU\_HAIM NURSE 400..500

...


Name	Type	Size	Data
NURSE_NAME	VARCHAR2	30	LastName + ' ' + FirstName
TELEPHONE_NUMBER	NUMBER		List('0526247728', '0533766598', '0525075779', '053148
NURSE_ID	NUMBER		Sequence(3600, 1, 4800)

SQL

Output

Statistics

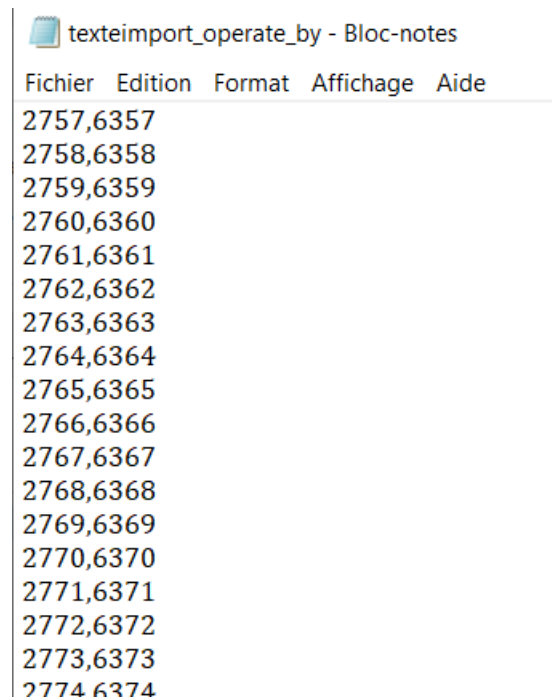
select \* from Nurse



	NURSE_NAME	TELEPHONE_NUMBER	NURSE_ID
1	Zevon Eric	533438067	3600
2	Brooke Moe	526247728	3601
3	Mahoney Mary-Louise	531480785	3602
4	Briscoe Ed	536089594	3603
5	Northam Rachael	531038589	3604
6	Azaria Joan	529149505	3605
7	Lizzy Sigourney	520778920	3606
8	Ronstadt Merle	529822770	3607
9	Puckett Edwin	525320641	3608
10	Slater Philip	535607196	3609
11	Mazar Victoria	521060217	3610
12	Gleeson Mena	521884551	3611
13	Spader Bo	533545787	3612
14	Richardson Azucar	520539639	3613
15	Farrell Doug	529093133	3614
16	McIntyre Peter	528359152	3615
17	Gallagher Millie	538923687	3616
18	Kennedy Terry	537810436	3617
19	Rawls Minnie	533426157	3618
20	Sweet Holly	531428850	3619
21	Ward First	522486321	3620

## Entering data by TEXT file:

Inserting data into the Operate\_by table:



textimport\_operate\_by - Bloc-notes

Fichier Edition Format Affichage Aide

2757,6357  
2758,6358  
2759,6359  
2760,6360  
2761,6361  
2762,6362  
2763,6363  
2764,6364  
2765,6365  
2766,6366  
2767,6367  
2768,6368  
2769,6369  
2770,6370  
2771,6371  
2772,6372  
2773,6373  
2774,6374

Data from Textfile
Data to Oracle

**File Data**
  
Doctor\_ID,Operation\_ID
  
2400,6000
  
2401,6001
  
2402,6002
  
2403,6003
  
2404,6004
  
2405,6005
  
2406,6006
  
2407,6007

**Configuration**

General

Fieldcount
  
2
  
☒ End at line-end
  
☒ Name in header
  
☒ Skip empty lines

Quote character
  
"
  
Comment line
  
  
Import lines
  
1 ..

Field1 (+0 .. ",") Doctor\_ID
  
Field2 (+0 .. ",") Operation\_ID

Field Start
  
☒ Relative position
  
☐ Absolute position
  
☐ Character
  
0
  
Field End
  
☐ Length
  
☒ Character
  
,

Filter

**Result Preview**

Doctor_ID	Operation_ID
2400	6000
2401	6001

Data from Textfile
Data to Oracle

**General**

Owner
  
C##ELIAOU\_HAIM
  
Commit every...
  
100

Table
  
OPERATE\_BY
  
☒ Overwrite duplicates
  
☐ Ignore duplicates

☐ Delete records
  
☐ Truncate table

Initializing Script
  
  
Finalizing Script

**Fields**
  
Field1 Doctor\_ID -> DOCTOR\_ID
  
Field2 Operation\_ID -> OPERATION\_ID

**Result Preview**









Doctor_ID	Operation_ID
2400	6000
2401	6001

SQL

Output

Statistics

select \* from Operate\_by

		DOCTOR_ID		OPERATION_ID	
	1	2400		6000	
	2	2401		6001	
	3	2402		6002	
	4	2403		6003	
▶	5	2404		6004	
	6	2405		6005	
	7	2406		6006	
	8	2407		6007	
	9	2408		6008	
	10	2409		6009	
	11	2410		6010	
	12	2411		6011	
	13	2412		6012	
	14	2413		6013	
	15	2414		6014	
	16	2415		6015	
	17	2416		6016	
	18	2417		6017	
	19	2418		6018	
	20	2419		6019	
	21	2420		6020	

Inserting data into the Assit\_by table:

Nurse\_ID,Operation\_ID

3600,6000

3601,6001

3602,6002

3603,6003

3604,6004

3605,6005

3606,6006

3607,6007

3608,6008

3609,6009

3610,6010

3611,6011

3612,6012

3613,6013

3614,6014

Data from Textfile Data to Oracle

### File Data

Nurse\_ID,Operation\_ID

3600,6000

3601,6001

3602,6002

3603,6003

3604,6004

3605,6005

3606,6006

3607,6007

### Configuration

#### General

Fieldcount

2

Quote character

"

☒ End at line-end

☒ Name in header

☒ Skip empty lines

Comment line

Import lines

1 ..

Field1 (+0 .. ",") Nurse\_ID

Field2 (+0 .. ",") Operation

Filter

### Result Preview

Nurse_ID	Operation_ID
3600	6000
3601	6001



SQL    Output    Statistics

```
select * from Assistby
```



		NURSE_ID	OPERATION_ID
▶	1	3600	6000
	2	3601	6001
	3	3602	6002
	4	3603	6003
	5	3604	6004
	6	3605	6005
	7	3606	6006
	8	3607	6007
	9	3608	6008
	10	3609	6009
	11	3610	6010
	12	3611	6011
	13	3612	6012
	14	3613	6013
	15	3614	6014
	16	3615	6015
	17	3616	6016
	18	3617	6017



Entering data by EXCEL (mockaroo) :

Entering data into the Operation table:

Operation_Date	Duration_Operation	Operation_ID	Patient_ID	Room_ID
03/03/2024	13	6000	100	1200
01/05/2024	10	6001	101	1201
01/27/2025	24	6002	102	1202
04/19/2024	15	6003	103	1203
12/16/2023	13	6004	104	1204
05/23/2024	2	6005	105	1205
11/13/2024	24	6006	106	1206
08/20/2024	23	6007	107	1207
09/12/2023	20	6008	108	1208
10/20/2023	21	6009	109	1209
01/28/2024	23	6010	110	1210
01/18/2025	15	6011	111	1211
01/23/2025	11	6012	112	1212
11/29/2024	23	6013	113	1213
08/06/2024	13	6014	114	1214

Data from TextfileData to Oracle

General

Owner

C##ELIAOU\_HAIM

Table

OPERATION

Commit every...

100

☒ Overwrite duplicates

☐ Ignore duplicates

☐ Delete records

☐ Truncate table

Initializing Script

Finalizing Script

Fields

Field1 Operation\_Date -> OPERATION\_DATE

Field2 Duration\_Operation -> DURATION\_OPERATION

Field3 Operation\_ID -> OPERATION\_ID

Field4 Patient\_ID -> PATIENT\_ID

Field5 Room\_ID -> ROOM\_ID

Field OPERATION\_DATE

Fieldtype Date

Create SQL

SQL function to\_date('MM/DD/YYYY')

additional Oracle processing, for example

Result Preview

Operation_Date	Duration_Operation	Operation_ID	Patient_ID	Room_ID
03/03/2024	13	6000	100	1200
01/05/2024	10	6001	101	1201
01/27/2025	24	6002	102	1202
04/19/2024	15	6003	103	1203
12/16/2023	13	6004	104	1204
05/23/2024	2	6005	105	1205
11/13/2024	24	6006	106	1206
08/20/2024	23	6007	107	1207

C##ELIAOU\_HAIM@VF

17/40501

465 records imported in 25.904 seconds (1 failed)

Data from Textfile
Data to Oracle

### File Data

```

Operation_Date,Duration_Operation,Operation_ID,Patient_ID,Room_ID
03/03/2024,13,6000,100,1200
01/05/2024,10,6001,101,1201
01/27/2025,24,6002,102,1202
04/19/2024,15,6003,103,1203
12/16/2023,13,6004,104,1204
05/23/2024,2,6005,105,1205

```

### Configuration

General

Fieldcount

☒ End at line-end  
☒ Name in header  
☒ Skip empty lines

Quote character  
"

Comment line

Import lines  
1 ..

Field1 (+0 .. ",") Operation\_Date  
Field2 (+0 .. ",") Duration\_Operation  
Field3 (+0 .. ",") Operation\_ID  
Field4 (+0 .. ",") Patient\_ID  
Field5 (+0 .. ",") Room\_ID

Field Start  
☒ Relative position   
☐ Absolute position  
☐ Character

Field End  
☐ Length  
☒ Character

Filter

### Result Preview

Operation_Date	Duration_Operation	Operation_ID	Patient_ID	Room_ID
03/03/2024	13	6000	100	1200
01/05/2024	10	6001	101	1201
01/27/2025	24	6002	102	1202
04/19/2024	15	6003	103	1203

```
select * from Operation
```

	OPERATION_DATE	DURATION_OPERATION	OPERATION_ID	PATIENT_ID	ROOM_ID
1	03/03/2024	13	6000	100	1200
2	05/01/2024	10	6001	101	1201
3	27/01/2025	24	6002	102	1202
4	19/04/2024	15	6003	103	1203
5	16/12/2023	13	6004	104	1204
6	23/05/2024	2	6005	105	1205
7	13/11/2024	24	6006	106	1206
8	20/08/2024	23	6007	107	1207
9	12/09/2023	20	6008	108	1208
10	20/10/2023	21	6009	109	1209
11	28/01/2024	23	6010	110	1210
12	18/01/2025	15	6011	111	1211
13	23/01/2025	11	6012	112	1212
14	29/11/2024	23	6013	113	1213
15	06/08/2024	2	6014	114	1214
16	15/01/2025	19	6015	115	1215
17	27/12/2024	1	6016	116	1216
18	16/10/2023	16	6017	117	1217
19	07/04/2024	21	6018	118	1218
20	09/11/2024	23	6019	119	1219

## Inserting data by INSERT commands:

SQL


Output

Statistics

INSERT INTO Operating\_Room (Room\_ID, Availability, Max\_number\_people) VALUES  
(2, 'Available', 5);  
select \* from Operating\_Room

Insert operating\_room

Select operating\_room



	ROOM_ID	AVAILABILITY	MAX_NUMBER_PEOPLE
▶	1	79 Yes	14
	2	71 Yes	20
	3	4 No	1
	4	67 Yes	6
	5	30 Yes	11
	6	15 No	8
	7	69 No	16
	8	43 No	8
	9	47 No	16
	10	52 Yes	20
	11	1 No	20

## חלק 2:

שאלות:

### Select:

1. List all operations performed in 2023, showing the patient name, doctor name, and operation duration. Order by operation date.
2. Retrieve the average operation duration for each doctor in a specific month ( June 2023) and list their specialties.
3. List all patients who had an operation in a room with more than 10 people capacity and show the illness and operation details.
4. Show the count of operations performed each month in 2024 along with the total duration of operations per month.

1.

```
SELECT
    p.Patient_Name,
    d.Doctor_Name,
    o.Operation_Date,
    o.Duration_Operation
FROM Operation o
JOIN Patient p ON o.Patient_ID = p.Patient_ID
JOIN Operate_by ob ON o.Operation_ID = ob.Operation_ID
JOIN Doctor d ON ob.Doctor_ID = d.Doctor_ID
WHERE EXTRACT(YEAR FROM o.Operation_Date) = 2023
ORDER BY o.Operation_Date;
```

	PATIENT_NAME	DOCTOR_NAME	OPERATION_DATE	DURATION_OPERATION
1	Alda Barbara	Rea Billy	13/06/2023	19
2	Evelt Tzi	McGinley Davey	15/06/2023	9
3	Dalley Mindy	Lennix Toshio	15/06/2023	17
4	Daniels Rowan	Hatosy Freddy	16/06/2023	7
5	Dooley Cyndi	Alda Kimberly	16/06/2023	12
6	Webb Miki	Macy Bette	17/06/2023	16
7	Olin Yaphet	Springfield Junior	17/06/2023	24
8	Conlee Jared	de Lancie Julia	20/06/2023	10
9	England Don	Neville Thelma	27/06/2023	10
10	Feuerstein Denzel	Hersh Sean	27/06/2023	3
11	Curry Samantha	Apple Javon	27/06/2023	7
12	Popper Terrence	Bracco Josh	30/06/2023	20
13	Crouch Helen	Soul Kevin	02/07/2023	16

2.

```

SELECT
    d.Doctor_Name,
    d.Specialty,
    AVG(o.Duration_Operation) AS Average_Operation_Duration
FROM Doctor d
JOIN Operate_by ob ON d.Doctor_ID = ob.Doctor_ID
JOIN Operation o ON ob.Operation_ID = o.Operation_ID
WHERE EXTRACT(YEAR FROM o.Operation_Date) = 2023
    AND EXTRACT(MONTH FROM o.Operation_Date) = 6
GROUP BY d.Doctor_Name, d.Specialty;

```

	DOCTOR_NAME	SPECIALTY	AVERAGE_OPERATION_DURATION
1	Rea Billy	Joint Replacement Surgery	19
2	Springfield Junior	Plastic Surgery	24
3	McGinley Davey	Neurointerventional Surgery	9
4	Hatosy Freddy	Endocrine Surgery	7
5	Neville Thelma	Orthopedic Surgery	10
6	Alda Kimberly	Gynecologic Surgery	12
7	de Lancie Julia	Otolaryngology	10
8	Hersh Sean	Pediatric Cardiothoracic Surgery	3
9	Apple Javon	Pediatric Trauma Surgery	7
10	Bracco Josh	General Surgery	20
11	Lennix Toshiro	Plastic Surgery	17
12	Macy Bette	Colorectal Surgery	16

3.

```

SELECT
    p.Patient_Name,
    p.Illness,
    o.Operation_ID,
    o.Operation_Date,
    r.Max_number_people
FROM Patient p
JOIN Operation o ON p.Patient_ID = o.Patient_ID
JOIN Operating_Room r ON o.Room_ID = r.Room_ID
WHERE r.Max_number_people > 10;

```

	PATIENT_NAME	ILLNESS	OPERATION_ID	OPERATION_DATE	MAX_NUMBER_PEOPLE
1	Place Beth	appendicitis	6328	19/07/2023	18
2	Birch Freddie	craniotomy	6329	28/05/2024	14
3	LaSalle Clea	hip replacement	6330	24/10/2023	11
4	Popper Terrence	adrenalectomy	6331	30/06/2023	16
5	Rosas Tori	melanoma	6333	02/02/2024	11
6	Bragg Kazem	coronary artery disease	6334	19/01/2024	13
7	Crouch Helen	bladder cancer	6336	02/07/2023	16
8	Isaacs Mac	cholecystectomy	6338	01/01/2024	14
9	Kramer Gordon	spinal fusion	6345	23/11/2024	16
10	Fehr Matt	cleft lip and palate	6346	07/11/2023	16
11	Curtis Adina	testicular cancer	6352	25/08/2024	15
12	Latifah Meryl	kidney cancer	6353	26/06/2024	11

4.

```

SELECT
    EXTRACT(MONTH FROM o.Operation_Date) AS Operation_Month,
    COUNT(o.Operation_ID) AS Total_Operations,
    SUM(o.Duration_Operation) AS Total_Duration
FROM Operation o
WHERE EXTRACT(YEAR FROM o.Operation_Date) = 2024
GROUP BY EXTRACT(MONTH FROM o.Operation_Date)
ORDER BY Operation_Month;

```

	OPERATION_MONTH	TOTAL_OPERATIONS	TOTAL_DURATION
1	1	21	243
2	2	32	395
3	3	16	182
4	4	25	308
5	5	18	192
6	6	19	232
7	7	28	307
8	8	24	332
9	9	19	254
10	10	19	277
11	11	26	311
12	12	21	261

### Delete:

1. Delete operations that were performed in a room with less than 5 people capacity and lasted more than 4 hours.
2. Delete all equipment that has not been used in any operation and is in 'not available' status.

1.

```

DELETE FROM Operation
WHERE Room_ID IN (
    SELECT Room_ID
    FROM Operating_Room
    WHERE Max_number_people < 5
) AND Duration_Operation > 240;

```

2.

```

DELETE FROM Equipement
WHERE Equipement_ID NOT IN (
    SELECT DISTINCT e.Equipement_ID
    FROM Equipement e
    JOIN Operating_Room r ON e.Room_ID = r.Room_ID
    JOIN Operation o ON r.Room_ID = o.Room_ID
) AND Equipement_Status = 'not available';

```

7:40 C##ELIAOU\_HAIM@XE [17:59:01] 1 row deleted in 0,015 seconds

### Update:

1. Update the status of all equipment in a specific room ( Room\_ID = 1415) to 'maintenance' if they were purchased before 2024-08-09.
2. Update the availability of operating rooms to 'available' if they have had no operations in the last 6 months

1.

```

UPDATE Equipement
SET Equipement_Status = 'maintenance'
WHERE Room_ID = 1415
AND Equipement_Purchase_Date < TO_DATE('2024-08-09', 'YYYY-MM-DD');

```

4:69 C##ELIAOU\_HAIM@XE [18:13:42] 3 rows updated in 0,004 seconds

2.

```

UPDATE Operating_Room
SET Availability = 'available'
WHERE Room_ID NOT IN (
    SELECT Room_ID
    FROM Operation
    WHERE Operation_Date > ADD_MONTHS(SYSDATE, -6)
);

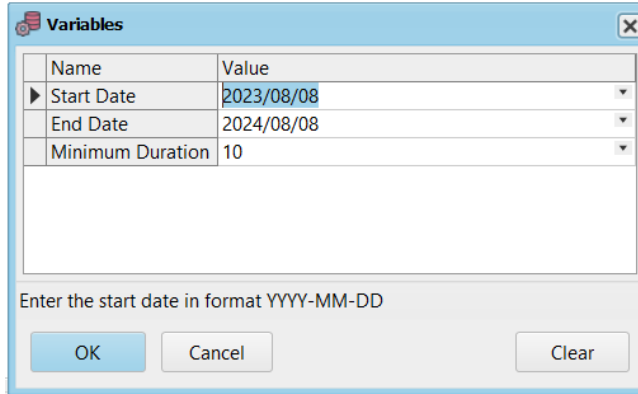
```

7:3 C##ELIAOU\_HAIM@XE [18:15:52] 148 rows updated in 0,059 seconds

## שאלות עם פרמטרים:

### ParamsQueries.sql:

#### 1. Operations between specific dates with a minimum duration



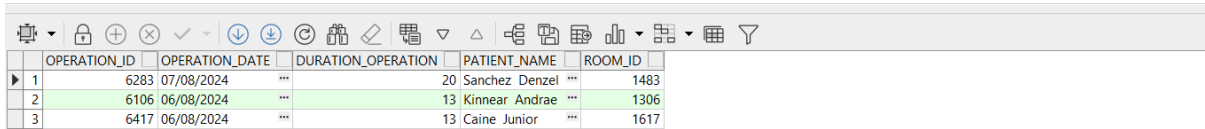
A dialog box titled "Variables" with a close button (X) in the top right corner. It contains a table with two columns: "Name" and "Value". The table has three rows: "Start Date" with value "2023/08/08", "End Date" with value "2024/08/08", and "Minimum Duration" with value "10". Below the table is a text input field with the placeholder text "Enter the start date in format YYYY-MM-DD". At the bottom are three buttons: "OK", "Cancel", and "Clear".

Name	Value
Start Date	2023/08/08
End Date	2024/08/08
Minimum Duration	10

Enter the start date in format YYYY-MM-DD

OK Cancel Clear

```
SELECT
    o.Operation_ID,
    o.Operation_Date,
    o.Duration_Operation,
    p.Patient_Name,
    r.Room_ID
FROM
    Operation o
JOIN
    Patient p ON o.Patient_ID = p.Patient_ID
JOIN
    Operating_Room r ON o.Room_ID = r.Room_ID
WHERE
    o.Operation_Date BETWEEN TO_DATE(&<name="Start Date" hint="Enter the start date in format YYYY-MM-DD" type="string">, 'YYYY-MM-DD')
    AND TO_DATE(&<name="End Date" hint="Enter the end date in format YYYY-MM-DD" type="string">, 'YYYY-MM-DD')
    AND o.Duration_Operation > &<name="Minimum Duration" hint="Enter the minimum duration in minutes" type="integer">
ORDER BY
    o.Operation_Date DESC;
```



A screenshot of a database query results window. It shows a table with 6 columns: OPERATION\_ID, OPERATION\_DATE, DURATION\_OPERATION, PATIENT\_NAME, and ROOM\_ID. The table has 3 rows of data. The first row is highlighted in blue, the second in green, and the third in white. Above the table is a toolbar with various icons for query execution and formatting.

	OPERATION_ID	OPERATION_DATE	DURATION_OPERATION	PATIENT_NAME	ROOM_ID
1	6283	07/08/2024	...	20 Sanchez Denzel	1483
2	6106	06/08/2024	...	13 Kinnear Andrae	1306
3	6417	06/08/2024	...	13 Caine Junior	1617

#### 2. Doctors with a specific specialty who performed more than a certain number of operations.



Variables

Name	Value
Specialty	Neurosurgery
Minimum Operations	0

OK

Cancel

Clear

```

SELECT
  d.Doctor_ID,
  d.Doctor_Name,
  d.Specialty,
  COUNT(ob.Operation_ID) AS NumberOfOperations
FROM
  Doctor d
JOIN
  Operate_by ob ON d.Doctor_ID = ob.Doctor_ID
WHERE
  d.Specialty = '<name="Specialty" list="select DISTINCT Specialty from Doctor order by Specialty">'
GROUP BY
  d.Doctor_ID, d.Doctor_Name, d.Specialty
HAVING
  COUNT(ob.Operation_ID) > <name="Minimum Operations" hint="Enter the minimum number of operations" type="integer">
ORDER BY
  NumberOfOperations DESC;

```

	DOCTOR_ID	DOCTOR_NAME	SPECIALTY	NUMBEROFOPERATIONS
1	2410	Hauer Anna	Neurosurgery	1
2	2465	Dafoe Nastassja	Neurosurgery	1
3	2490	Chan Dabney	Neurosurgery	1
4	2556	Danger Neil	Neurosurgery	1
5	2770	Marin Jann	Neurosurgery	1
6	2564	McGoohan Emm	Neurosurgery	1
7	2677	Lorenz Christmas	Neurosurgery	1
8	2756	Holmes Lydia	Neurosurgery	1
9	2559	Culkin Mary Beth	Neurosurgery	1

3. Nurses with a minimum number of operations assisted in a specific year.

Variables

Name	Value
Year	2023
Minimum Operations	0

OK

Cancel

Clear

Enter the year

```

SELECT
    n.Nurse_ID,
    n.Nurse_Name,
    n.Telephone_number,
    COUNT(ab.Operation_ID) AS NumberOfOperations
FROM
    Nurse n
JOIN
    Assist_by ab ON n.Nurse_ID = ab.Nurse_ID
JOIN
    Operation o ON ab.Operation_ID = o.Operation_ID
WHERE
    EXTRACT(YEAR FROM o.Operation_Date) = &name="Year" hint="Enter the year" type="integer">
GROUP BY
    n.Nurse_ID, n.Nurse_Name, n.Telephone_number
HAVING
    COUNT(ab.Operation_ID) > &name="Minimum Operations" hint="Enter the minimum number of operations" type="integer">
ORDER BY
    NumberOfOperations DESC;|

```

	NURSE_ID	NURSE_NAME	TELEPHONE_NUMBER	NUMBEROFOPERATIONS
1	3604	Northam Rachael	531038589	1
2	3608	Puckett Edwin	525320641	1
3	3609	Slater Philip	535607196	1
4	3617	Kennedy Terry	537810436	1
5	3620	Ward First	522486321	1
6	3621	Lipnicki Rosario	521259541	1
7	3625	Davies Giovanni	531453475	1

4. Equipment purchased before a specific date in rooms with a certain capacity.

Variables

Name	Value
Purchase Date	2024/08/08
Room Capacity	5

Enter the purchase date in format YYYY-MM-DD

OK

Cancel

Clear

```

SELECT
    e.Equipment_ID,
    e.Equipment_Name,
    e.Equipment_Purchase_Date,
    r.Room_ID,
    r.Max_number_people
FROM
    Equipment e
JOIN
    Operating_Room r ON e.Room_ID = r.Room_ID
WHERE
    e.Equipment_Purchase_Date < TO_DATE(&name="Purchase Date" hint="Enter the purchase date in format YYYY-MM-DD" type="string">, 'YYYY-MM-DD')
    AND r.Max_number_people >= &name="Room Capacity" hint="Enter the minimum room capacity" type="integer">
ORDER BY
    e.Equipment_Purchase_Date DESC;|

```

	EQUIPMENT_ID	EQUIPMENT_NAME	EQUIPMENT_PURCHASE_DATE	ROOM_ID	MAX_NUMBER_PEOPLE
1	4813	Surgical tray	28/07/2024	1274	20
2	5168	Surgical tray	28/07/2024	1327	16
3	5121	Cautery pencil	28/07/2024	1550	8
4	4988	Surgical microscope	28/07/2024	1356	5
5	4987	Retractor	28/07/2024	1380	9
6	4973	Surgical lights	28/07/2024	1560	10
7	4903	Endoscope	28/07/2024	1503	7
8	5270	Anesthesia machine	28/07/2024	1362	16
9	4964	IV pole	27/07/2024	1442	7
10	4810	Foot pedal	27/07/2024	1372	12
11	5158	Surgical gloves	27/07/2024	1479	13

**Constraints.sql:**

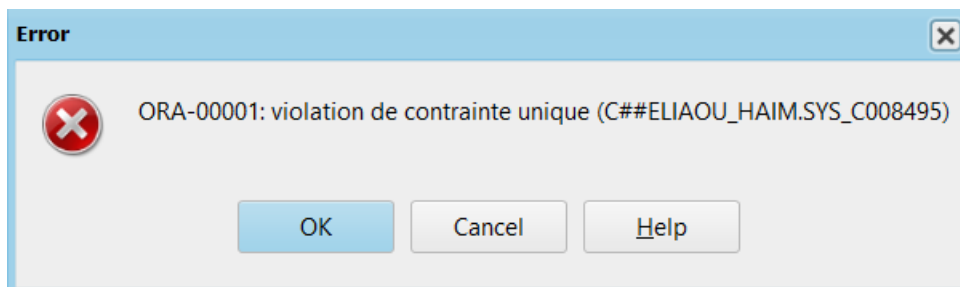
## 1. Adding CHECK constraints on Operation.

```
ALTER TABLE Operation
ADD CONSTRAINT chk_duration_operation CHECK (Duration_Operation > 0),
ADD CONSTRAINT chk_operation_id CHECK (Operation_ID > 0);

-- Insert statement to test the constraint
INSERT INTO Operation (Operation_ID, Operation_Date, Duration_Operation, Patient_ID, Room_ID)
VALUES (1, TO_DATE('2024-04-27', 'YYYY-MM-DD'), 0, 1, 101); -- This will fail due to Duration_Operation check constraint

-- Insert a valid record
INSERT INTO Operation (Operation_ID, Operation_Date, Duration_Operation, Patient_ID, Room_ID)
VALUES (6002, TO_DATE('2024-04-27', 'YYYY-MM-DD'), 5, 434, 1203);

-- Select statement to verify the insert
SELECT Operation_ID, Duration_Operation
FROM Operation
WHERE Operation_ID = 6002;
```



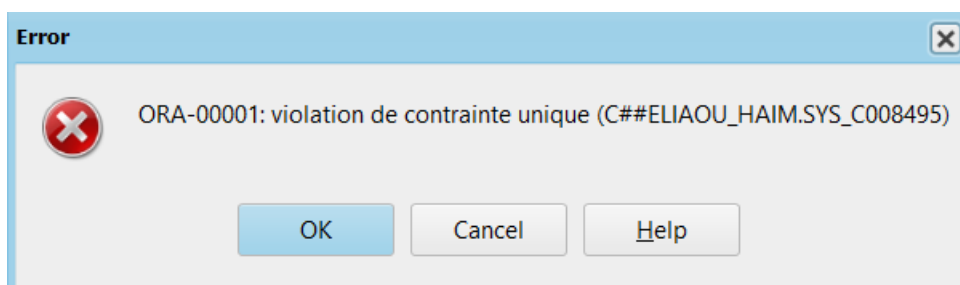
## 2. Adding UNIQUE constraint on Doctor

```
ALTER TABLE Doctor
ADD CONSTRAINT uniq_doctor_name UNIQUE (Doctor_Name);

-- Insert statement to test the constraint
INSERT INTO Doctor (Doctor_ID, Doctor_Name, Specialty)
VALUES (2410, 'Hauer Anna', 'Cardiology');

-- Attempt to insert a duplicate doctor name, which should fail
INSERT INTO Doctor (Doctor_ID, Doctor_Name, Specialty)
VALUES (2410, 'Hauer Anna', 'Neurosurgery'); -- This will fail due to the UNIQUE constraint

-- Select statement to verify the unique constraint
SELECT Doctor_Name
FROM Doctor;
```

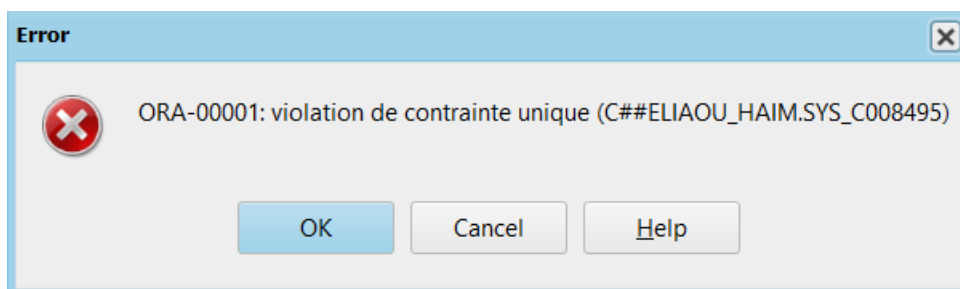


### 3. Adding DEFAULT constraint on Operating\_Room

```
ALTER TABLE Operating_Room
MODIFY Availability DEFAULT 'available';

-- Insert statement to test the default value
INSERT INTO Operating_Room (Room_ID, Max_number_people)
VALUES (1274, 4);

-- Select statement to verify the default value
SELECT Room_ID, Availability
FROM Operating_Room
WHERE Room_ID = 1274;
```



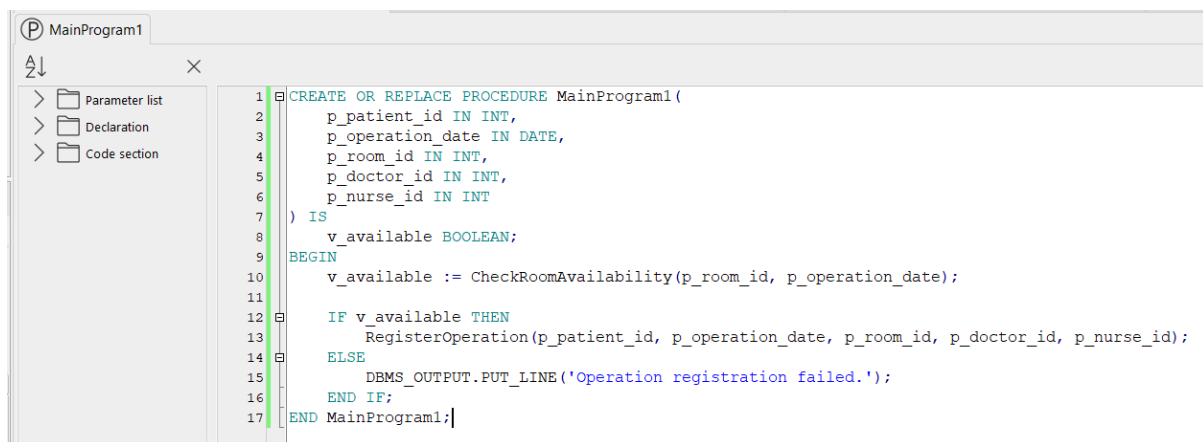
**חלק 3:**

תוכנית א:

Program description:

Record a complete operation. This main program calls the CheckRoomAvailability function and the RegisterOperation procedure. We check room availability and register an operation if a room is available; otherwise, it notifies about registration failure

The main program:



```
1 CREATE OR REPLACE PROCEDURE MainProgram1(  
2     p_patient_id IN INT,  
3     p_operation_date IN DATE,  
4     p_room_id IN INT,  
5     p_doctor_id IN INT,  
6     p_nurse_id IN INT  
7 ) IS  
8     v_available BOOLEAN;  
9 BEGIN  
10    v_available := CheckRoomAvailability(p_room_id, p_operation_date);  
11  
12    IF v_available THEN  
13        RegisterOperation(p_patient_id, p_operation_date, p_room_id, p_doctor_id, p_nurse_id);  
14    ELSE  
15        DBMS_OUTPUT.PUT_LINE('Operation registration failed.');16    END IF;  
17 END MainProgram1;
```

A function checks if an operating room is available on a given date:

⌵

CheckRoomAvailability

Parameter list  
Declaration  
**Code section**  
Exception handler

Code section

Update

```
1 CREATE OR REPLACE FUNCTION CheckRoomAvailability(  
2     p_room_id IN INT,  
3     p_date IN DATE  
4 ) RETURN BOOLEAN IS  
5     v_available VARCHAR2(10);  
6 BEGIN  
7     -- Check if the room is available on the given date  
8     SELECT Availability INTO v_available  
9     FROM Operating_Room  
10    WHERE Room_ID = p_room_id;  
11  
12    -- If no room found, return false  
13    IF SQL%NOTFOUND THEN  
14        RETURN FALSE;  
15    END IF;  
16  
17    -- If room is already booked, return false  
18    IF v_available <> 'Yes' THEN  
19        RETURN FALSE;  
20    END IF;  
21  
22    -- Attempt to book the room for the given date  
23    BEGIN  
24        UPDATE Operating_Room  
25        SET Availability = 'BOOKED'  
26        WHERE Room_ID = p_room_id  
27            AND Availability = 'Yes'  
28            AND ROWNUM = 1; -- ROWNUM condition for single row update  
29  
30        IF SQL%ROWCOUNT = 1 THEN  
31            RETURN TRUE; -- Room successfully booked  
32        ELSE  
33            RETURN FALSE; -- Room could not be booked (probably already booked)  
34        END IF;  
35    END IF;  
36    EXCEPTION  
37    WHEN NO_DATA_FOUND THEN  
38        RETURN FALSE; -- No room found with given ID  
39    WHEN OTHERS THEN  
40        RETURN FALSE; -- Other unexpected errors  
41    END;  
42    EXCEPTION  
43    WHEN NO_DATA_FOUND THEN  
44        RETURN FALSE; -- No room found with given ID  
45    WHEN OTHERS THEN  
46        RETURN FALSE; -- Other unexpected errors  
47 END CheckRoomAvailability;
```

A procedure that records a new operation in the Operation table:

RegisterOperation

Parameter list

Declaration

Code section

Exception handler

Code section

Comment

```
1 CREATE OR REPLACE PROCEDURE RegisterOperation(  
2     p_patient_id IN INT,  
3     p_operation_date IN DATE,  
4     p_room_id IN INT,  
5     p_doctor_id IN INT,  
6     p_nurse_id IN INT  
7 ) IS  
8     v_operation_id INT;  
9 BEGIN  
10    -- We using an exception to handle missing data errors  
11    BEGIN  
12        -- Check if operation already exists for this patient on specified date  
13        SELECT Operation_ID INTO v_operation_id  
14        FROM Operation  
15        WHERE Patient_ID = p_patient_id  
16              AND Operation_Date = p_operation_date;  
17  
18        -- If operation for this patient on this date already exists  
19        RAISE_APPLICATION_ERROR(-20002, 'Operation already registered for this patient on specified date.');
```

20  
21 EXCEPTION  
22 WHEN NO\_DATA\_FOUND THEN  
23 NULL; -- No existing operation, proceed to registration  
24  
25 END;  
26  
27 -- We use DML to insert the operation  
28 INSERT INTO Operation(Operation\_Date, Duration\_Operation, Patient\_ID, Room\_ID)  
29 VALUES (p\_operation\_date, 3, p\_patient\_id, p\_room\_id);  
30  
31 -- We use a procedure to register the responsible doctor and nurse  
32 RegisterDoctorAndNurse(p\_doctor\_id, p\_nurse\_id, v\_operation\_id);  
33  
34 COMMIT;  
35 DBMS\_OUTPUT.PUT\_LINE('Operation successfully registered for Patient ' || p\_patient\_id || ' on ' || TO\_CHAR(p\_operation\_date,

36  
37 EXCEPTION  
38 WHEN OTHERS THEN  
39 DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);  
40 ROLLBACK;  
41 END RegisterOperation;

## Running the program

The data before the test:

```
SELECT *  
FROM Operating_Room  
WHERE Room_ID = 1615;
```

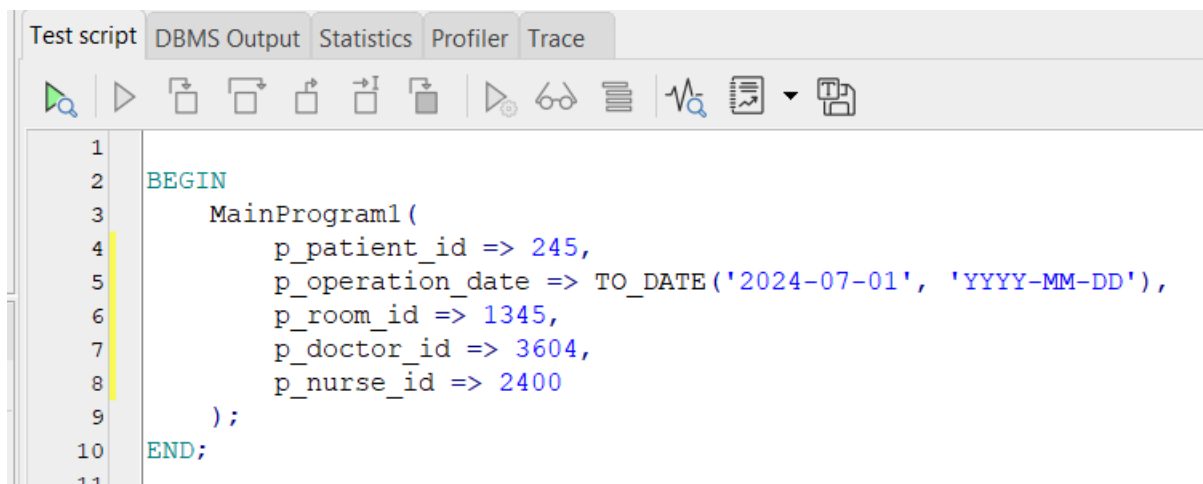
	ROOM_ID	AVAILABILITY	MAX_NUMBER_PEOPLE
▶ 1	1615	Yes	17

AVAILABILITY = Yes

```
SELECT *
FROM Operation
WHERE Patient_ID = 245;
```

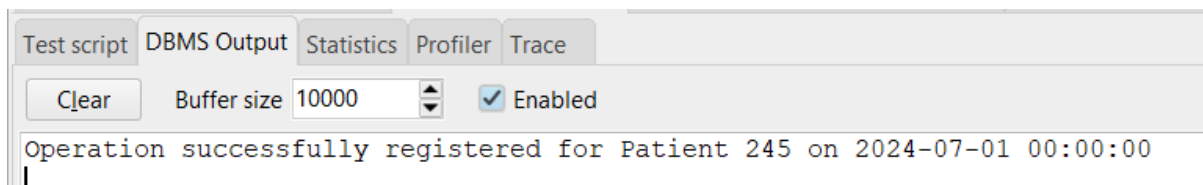
Juste one operation for the patient 245

After the test:



The screenshot shows a database test script editor with tabs for 'Test script', 'DBMS Output', 'Statistics', 'Profiler', and 'Trace'. The 'Test script' tab is active, displaying a PL/SQL procedure call. The code is as follows:

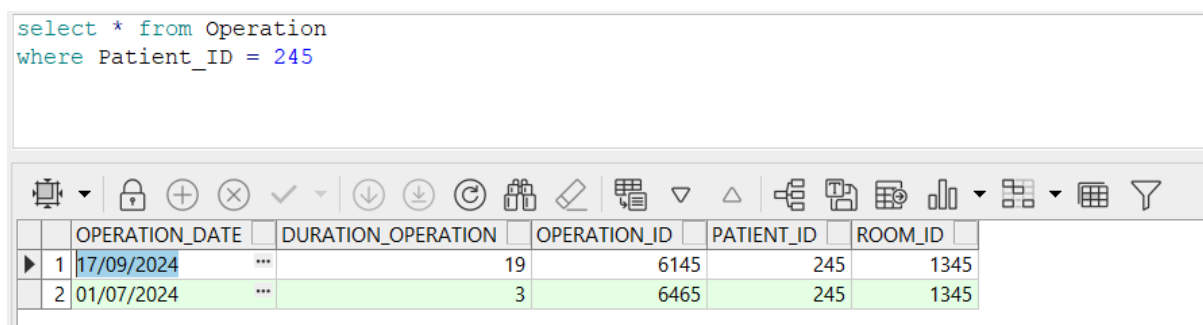
```
1  
2 BEGIN  
3     MainProgram1(  
4         p_patient_id => 245,  
5         p_operation_date => TO_DATE('2024-07-01', 'YYYY-MM-DD'),  
6         p_room_id => 1345,  
7         p_doctor_id => 3604,  
8         p_nurse_id => 2400  
9     );  
10 END;  
11
```



The screenshot shows the 'DBMS Output' tab in the database interface. It includes a 'Clear' button, a 'Buffer size' of 10000, and a checkbox for 'Enabled' which is checked. The output text reads:

```
Operation successfully registered for Patient 245 on 2024-07-01 00:00:00
```

The database after the test:



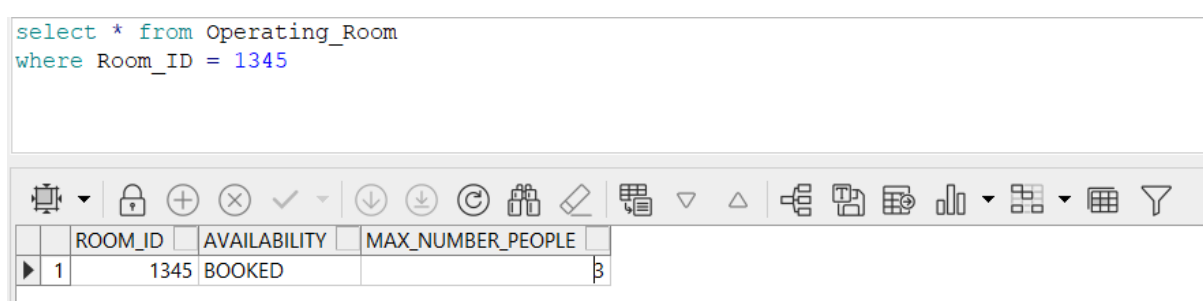
The screenshot shows a database query result for the 'Operation' table. The query is:

```
select * from Operation  
where Patient_ID = 245
```

The result is displayed in a table with the following columns: OPERATION\_DATE, DURATION\_OPERATION, OPERATION\_ID, PATIENT\_ID, and ROOM\_ID. The data is as follows:

	OPERATION_DATE	DURATION_OPERATION	OPERATION_ID	PATIENT_ID	ROOM_ID
1	17/09/2024	...	19	245	1345
2	01/07/2024	...	3	245	1345

Now we are tow operation for the patient 245.



The screenshot shows a database query result for the 'Operating\_Room' table. The query is:

```
select * from Operating_Room  
where Room_ID = 1345
```

The result is displayed in a table with the following columns: ROOM\_ID, AVAILABILITY, and MAX\_NUMBER\_PEOPLE. The data is as follows:

	ROOM_ID	AVAILABILITY	MAX_NUMBER_PEOPLE
1	1345	BOOKED	3

Now the AVAILABILITY = BOOKED.



### Program description:

This procedure checks the current status of an equipment identified by `p_equipment_id` and updates its status to `p_new_status` if the current status is 'AVAILABLE'

### The main program:

The screenshot shows a PL/SQL editor window titled 'MainProgram2'. On the left, there is a tree view with 'Parameter list', 'Declaration', and 'Code section'. The main area displays the following code:

```

1 CREATE OR REPLACE PROCEDURE MainProgram2(
2   p_equipment_id IN INT,
3   p_new_status IN VARCHAR2
4 ) IS
5   v_current_status VARCHAR2(30);
6 BEGIN
7   v_current_status := CheckEquipmentStatus(p_equipment_id);
8
9   IF v_current_status = 'available' THEN
10    UpdateEquipmentStatus(p_equipment_id, p_new_status);
11  ELSE
12    DBMS_OUTPUT.PUT_LINE('Equipment ' || p_equipment_id || ' is not available to update.');

```

A function retrieves the current status of an equipment identified by `p_equipment_id`. If the equipment is not found (NO\_DATA\_FOUND), it returns 'NOT FOUND'. For any other errors (OTHERS), it returns 'ERROR'.

The screenshot shows a PL/SQL editor window titled 'CheckEquipmentStatus'. On the left, there is a tree view with 'Parameter list', 'Declaration', 'Code section', and 'Exception handler'. The 'Exception handler' tab is selected. The main area displays the following code:

```

1 CREATE OR REPLACE FUNCTION CheckEquipmentStatus(
2   p_equipment_id IN INT
3 ) RETURN VARCHAR2 IS
4   v_status VARCHAR2(30);
5 BEGIN
6   SELECT Equipment_Status INTO v_status
7   FROM Equipment
8   WHERE Equipment_ID = p_equipment_id;
9
10  RETURN v_status;
11 EXCEPTION
12  WHEN NO_DATA_FOUND THEN
13    RETURN 'NOT FOUND';
14  WHEN OTHERS THEN
15    RETURN 'ERROR';
16 END CheckEquipmentStatus;
```

A procedure that updates the status of an equipment identified by `p_equipment_id` to `p_new_status`.

```
UpdateEquipmentStatus
Code section Update
1 CREATE OR REPLACE PROCEDURE UpdateEquipmentStatus(
2   p_equipment_id IN INT,
3   p_new_status IN VARCHAR2
4 ) IS
5 BEGIN
6   UPDATE Equipment
7   SET Equipment_Status = p_new_status
8   WHERE Equipment_ID = p_equipment_id;
9
10  DBMS_OUTPUT.PUT_LINE('Equipment ' || p_equipment_id || ' status updated to ' || p_new_status);
11 EXCEPTION
12 WHEN OTHERS THEN
13   DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
14 END UpdateEquipmentStatus;
```

Running the program

The data before the test:


SQL Output Statistics						
<pre>select * from Equipment where equipment_id = 4807</pre>						
	EQUIPEMENT_ID	EQUIPEMENT_PURCHASE_DATE	EQUIPEMENT_NAME	EQUIPEMENT_STATUS	ROOM_ID	
1	4807	23/03/2024	Orthopedic implants	available	1652	

Running the program

The data before the test:

Test script DBMS Output Statistics Profiler Trace				
Clear	Buffer size	10000	Enabled	
Equipment 4807 status updated to IN USE				

Test scriptDBMS OutputStatisticsProfilerTrace



123456

```
BEGIN
    MainProgram2(
        p_equipment_id => 4807,
        p_new_status => 'IN USE'
    );
END;
```