בס"ד

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. Equipement (צִיוּד):

- 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 (אָחוֹת):

- 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.

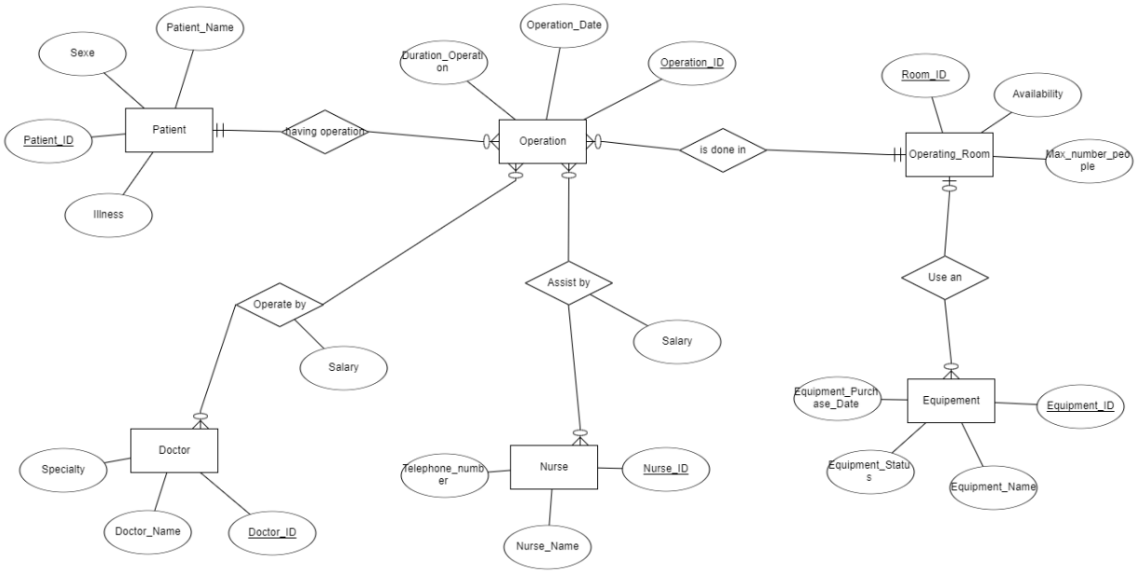
An operation takes place in a single room. (M: 1)
A room can accommodate several operations.

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

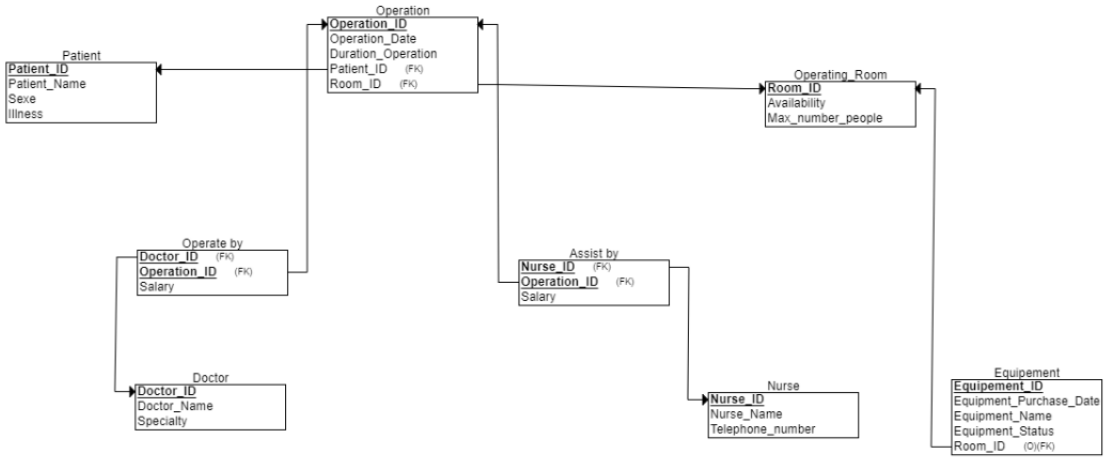
One piece of equipment can be used by a maximum of one operations room. (1: N)
An operating room may require several pieces of equipment.

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

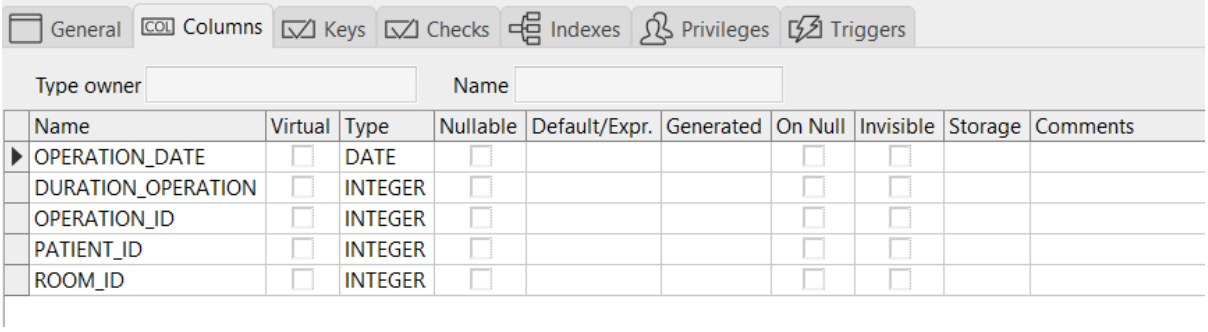
Name	Type	Columns	Enabled	Referencing table	Referencing columns	On Delete	Deferrable	Deferred	Validated	Last change
SYS_C008318	Primary	PATIENT_ID	☑				☐	☐	☑	27/05/2024 15:14:36

General | Columns | Keys | Checks | Indexes | Privileges | Triggers

Type owner | Name

Name	Virtual	Type	Nullable	Default/Expr.	Generated	On Null	Invisible	Storage	Comments
PATIENT_ID	☐	INTEGER	☐			☐	☐		
SEXE	☐	VARCHAR2(30)	☐			☐	☐		
PATIENT_NAME	☐	VARCHAR2(30)	☐			☐	☐		
ILLNESS	☐	VARCHAR2(100)	☐			☐	☐		

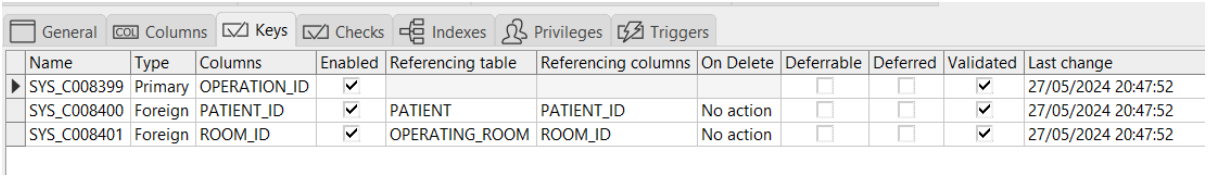
Creating the **Operation** table:



General | Columns | Keys | Checks | Indexes | Privileges | Triggers

Type owner Name

Name	Virtual	Type	Nullable	Default/Expr.	Generated	On Null	Invisible	Storage	Comments
OPERATION_DATE		DATE							
DURATION_OPERATION		INTEGER							
OPERATION_ID		INTEGER							
PATIENT_ID		INTEGER							
ROOM_ID		INTEGER							

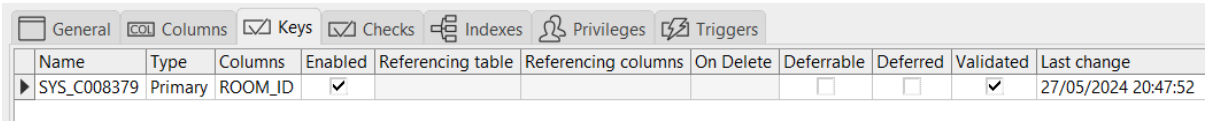


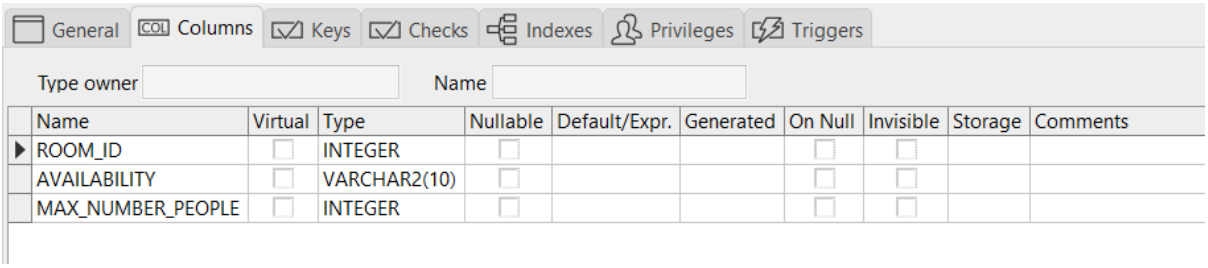
General | Columns | Keys | Checks | Indexes | Privileges | Triggers

Name	Type	Columns	Enabled	Referencing table	Referencing columns	On Delete	Deferrable	Deferred	Validated	Last change
SYS_C008399	Primary	OPERATION_ID	✓						✓	27/05/2024 20:47:52
SYS_C008400	Foreign	PATIENT_ID	✓	PATIENT	PATIENT_ID	No action			✓	27/05/2024 20:47:52
SYS_C008401	Foreign	ROOM_ID	✓	OPERATING_ROOM	ROOM_ID	No action			✓	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
```

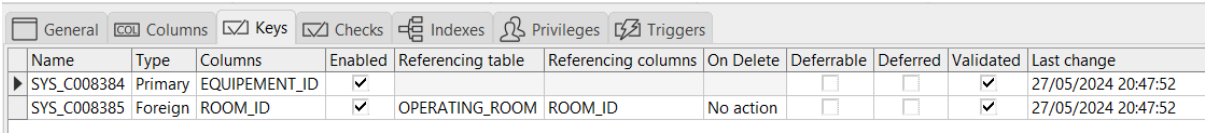
Creating the **Operating Room** table:





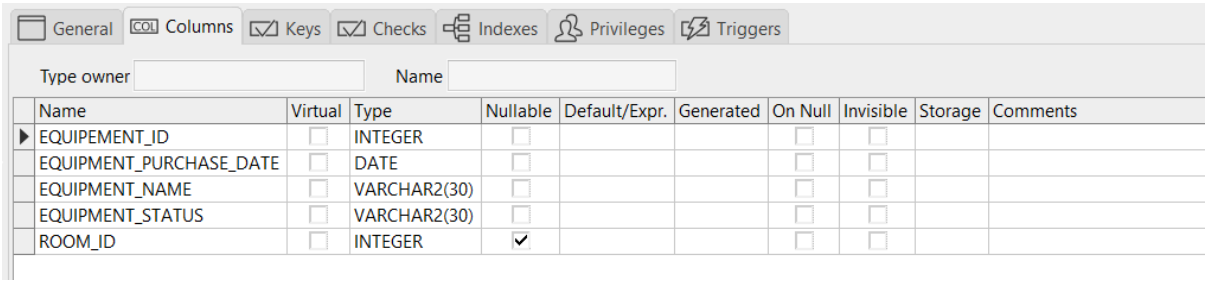

```
-- Create table
create table OPERATING_ROOM
(
  room_id            INTEGER not null,
  availability       VARCHAR2(10) not null,
  max_number_people INTEGER not null
)
tablespace SYSTEM
  pctfree 10
  pctused 40
  initrans 1
  maxtrans 255
  storage
  (
    initial 64K
    next 1M
    minextents 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	✓						✓	27/05/2024 20:47:52
SYS_C008385	Foreign	ROOM_ID	✓	OPERATING_ROOM	ROOM_ID	No action			✓	27/05/2024 20:47:52



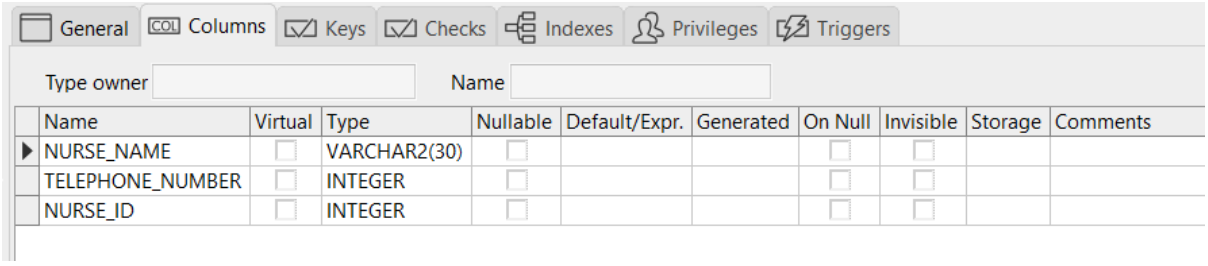
General | Columns | Keys | Checks | Indexes | Privileges | Triggers

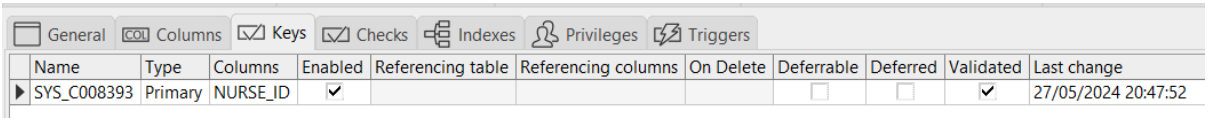
Type owner Name

Name	Virtual	Type	Nullable	Default/Expr.	Generated	On Null	Invisible	Storage	Comments
EQUIPEMENT_ID		INTEGER							
EQUIPMENT_PURCHASE_DATE		DATE							
EQUIPMENT_NAME		VARCHAR2(30)							
EQUIPMENT_STATUS		VARCHAR2(30)							
ROOM_ID		INTEGER	✓						

```
-- Create table
create table EQUIPEMENT
(
  equipement_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
```

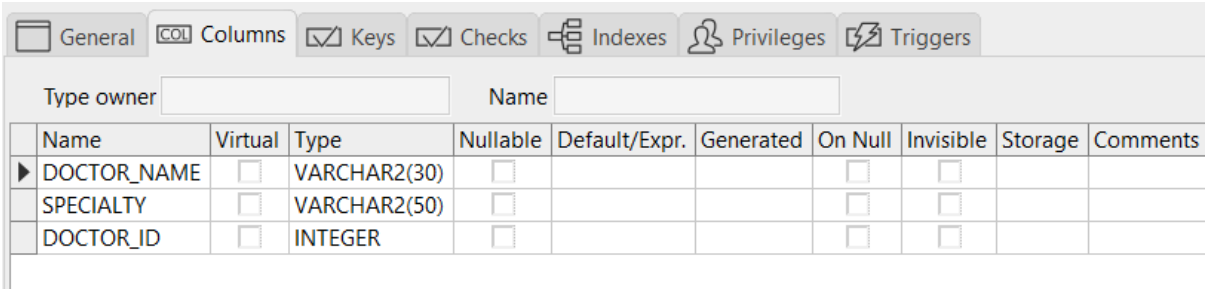
Creating the **Nurse** table:

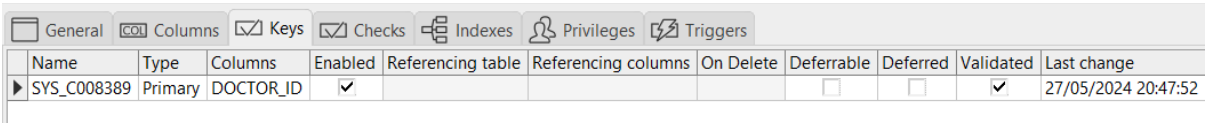





```
-- Create table
create table NURSE
(
  nurse_name       VARCHAR2(30) not null,
  telephone_number INTEGER not null,
  nurse_id         INTEGER not null
)
tablespace SYSTEM
  pctfree 10
  pctused 40
  initrans 1
  maxtrans 255
  storage
  (
    initial 64K
    next 1M
    minextents 1
    maxextents 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
```

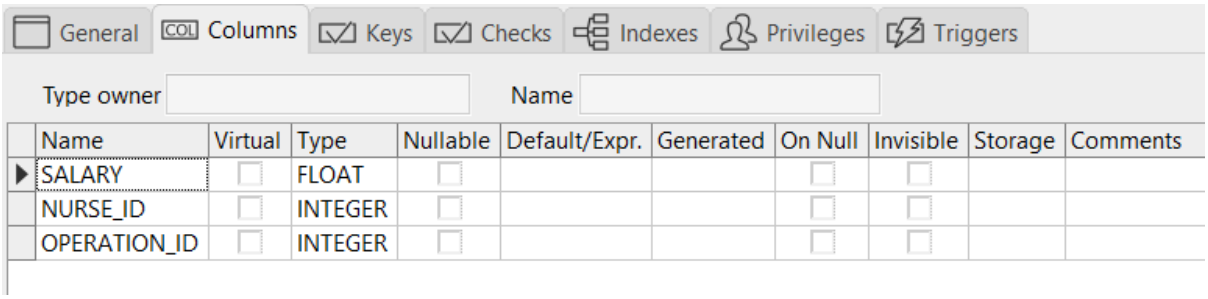
Creating the **Doctor** table:






```
-- 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
```

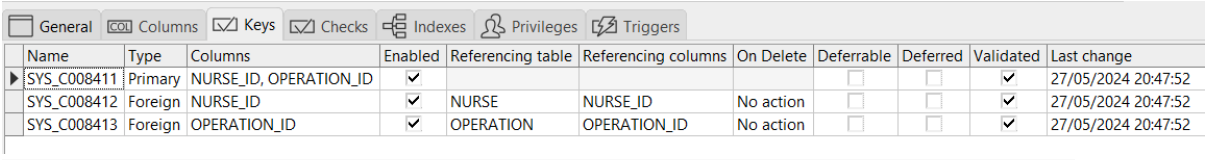
Creating the **Assist by** table:



General | Columns | Keys | Checks | Indexes | Privileges | Triggers

Type owner | Name

Name	Virtual	Type	Nullable	Default/Expr.	Generated	On Null	Invisible	Storage	Comments
SALARY		FLOAT							
NURSE_ID		INTEGER							
OPERATION_ID		INTEGER							

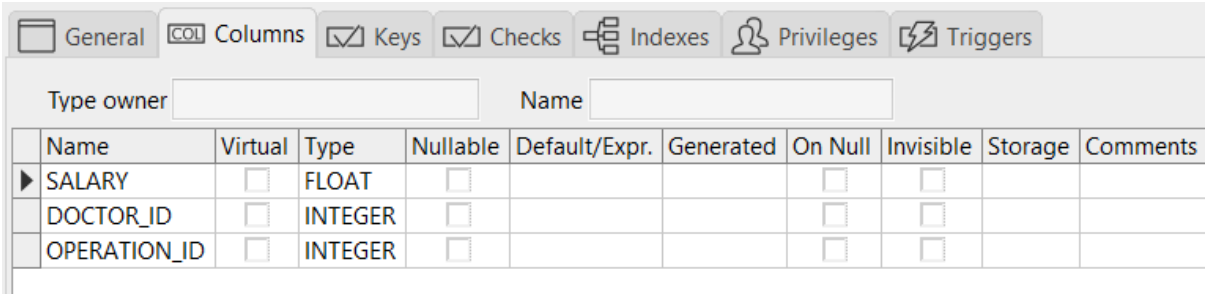


General | Columns | Keys | Checks | Indexes | Privileges | Triggers

Name	Type	Columns	Enabled	Referencing table	Referencing columns	On Delete	Deferrable	Deferred	Validated	Last change
SYS_C008411	Primary	NURSE_ID, OPERATION_ID	✓						✓	27/05/2024 20:47:52
SYS_C008412	Foreign	NURSE_ID	✓	NURSE	NURSE_ID	No action			✓	27/05/2024 20:47:52
SYS_C008413	Foreign	OPERATION_ID	✓	OPERATION	OPERATION_ID	No action			✓	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
```

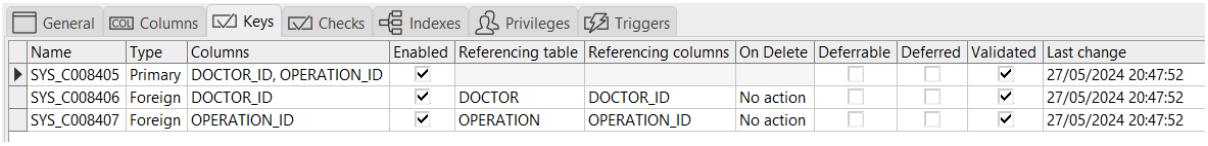
Creating the **Operate by** table:



General | Columns | Keys | Checks | Indexes | Privileges | Triggers

Type owner | Name

Name	Virtual	Type	Nullable	Default/Expr.	Generated	On Null	Invisible	Storage	Comments
SALARY	☐	FLOAT	☐			☐	☐		
DOCTOR_ID	☐	INTEGER	☐			☐	☐		
OPERATION_ID	☐	INTEGER	☐			☐	☐		



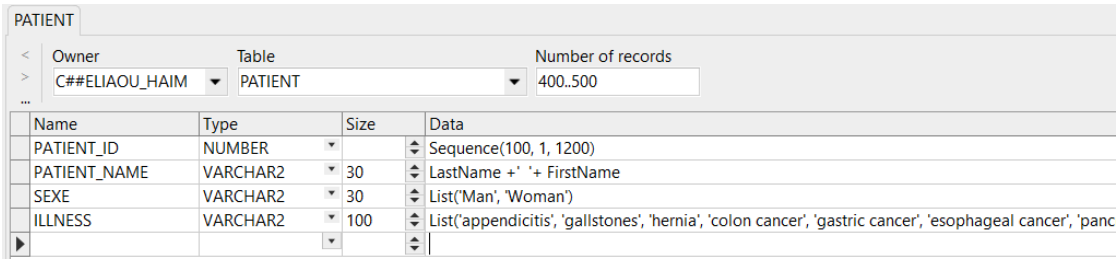
General | Columns | Keys | Checks | Indexes | Privileges | Triggers

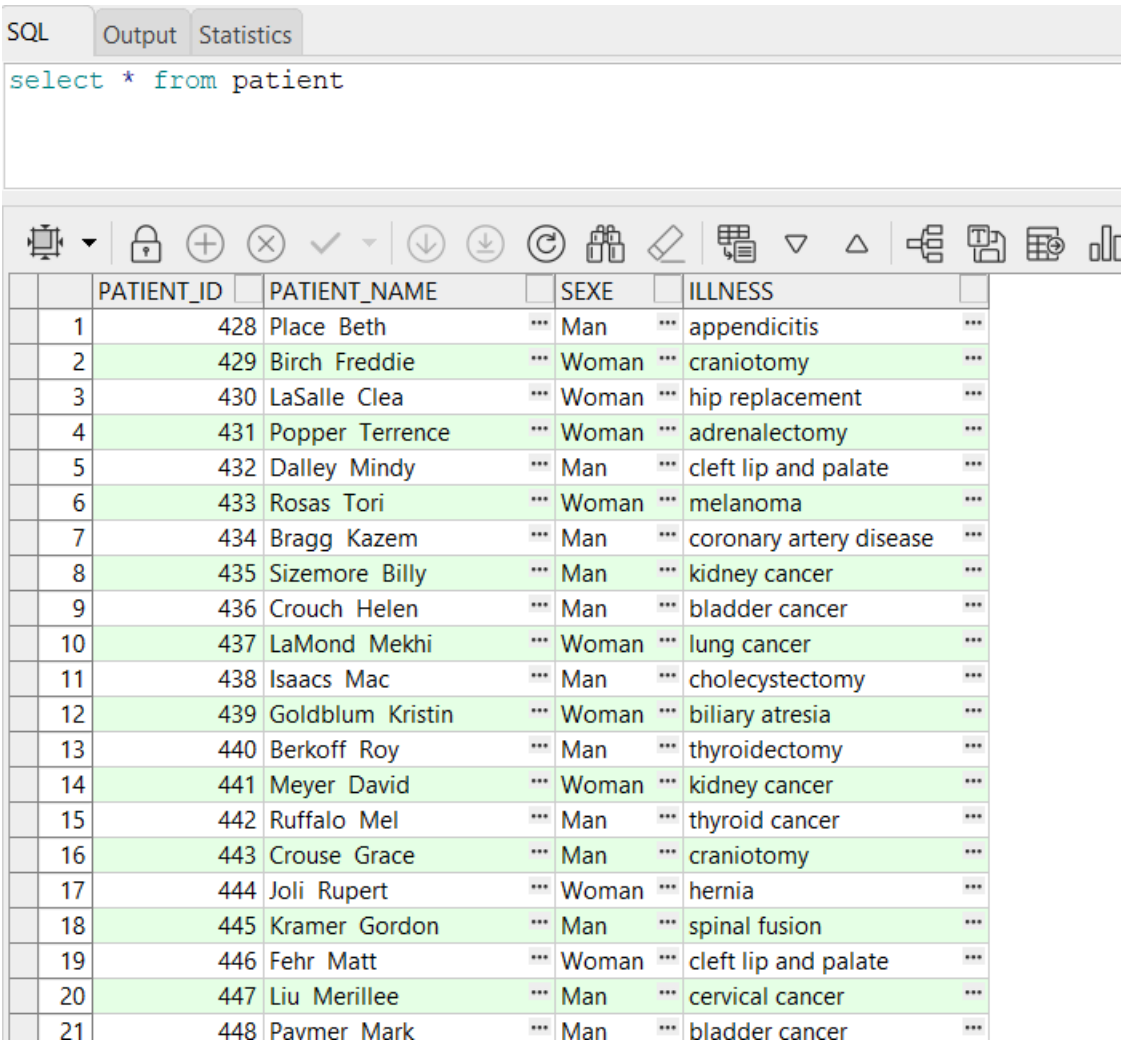
Name	Type	Columns	Enabled	Referencing table	Referencing columns	On Delete	Deferrable	Deferred	Validated	Last change
SYS_C008405	Primary	DOCTOR_ID, OPERATION_ID	☑				☐	☐	☑	27/05/2024 20:47:52
SYS_C008406	Foreign	DOCTOR_ID	☑	DOCTOR	DOCTOR_ID	No action	☐	☐	☑	27/05/2024 20:47:52
SYS_C008407	Foreign	OPERATION_ID	☑	OPERATION	OPERATION_ID	No action	☐	☐	☑	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.

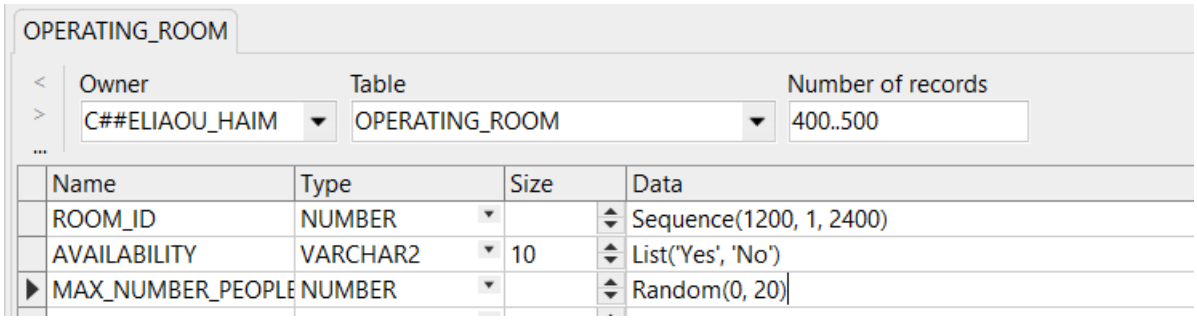
Entering data into the Patient table:

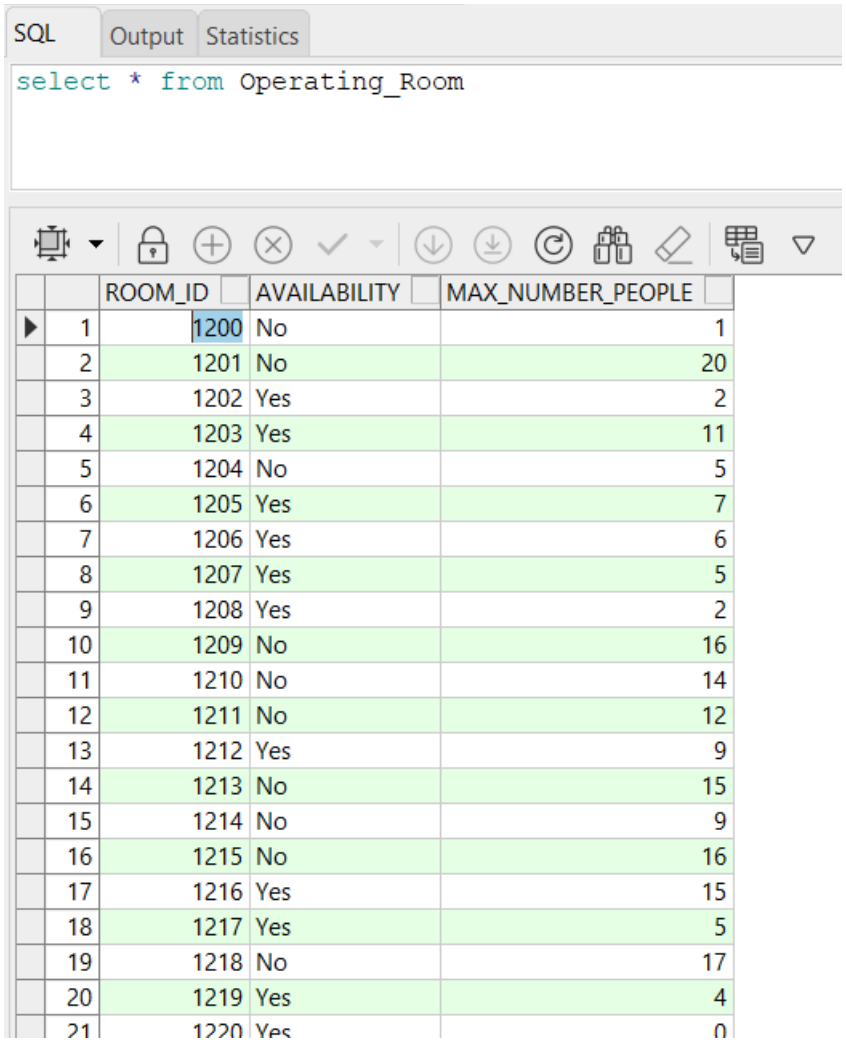




	PATIENT_ID	PATIENT_NAME	SEXE	ILLNESS
1	428	Place Beth	Man	appendicitis
2	429	Birch Freddie	Woman	craniotomy
3	430	LaSalle Clea	Woman	hip replacement
4	431	Popper Terrence	Woman	adrenalectomy
5	432	Dalley Mindy	Man	cleft lip and palate
6	433	Rosas Tori	Woman	melanoma
7	434	Bragg Kazem	Man	coronary artery 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	Paymer Mark	Man	bladder cancer

Entering data into the Operating_Room table:





	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: C##ELIAOU_HAIM | Table: EQUIPEMENT | Number of records: 400..500

Name	Type	Size	Data
EQUIPEMENT_ID	NUMBER		Sequence(4800, 1, 6000)
EQUIPMENT_PURCHAS	DATE		List('01/01/2024', '14/01/2024', '28/01/2024', '05/02/2024', '19/02/202
EQUIPMENT_NAME	VARCHAR2	30	List('Scalpel', 'Forceps', 'Hemostat', 'Surgical scissors', 'Needle holder',
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

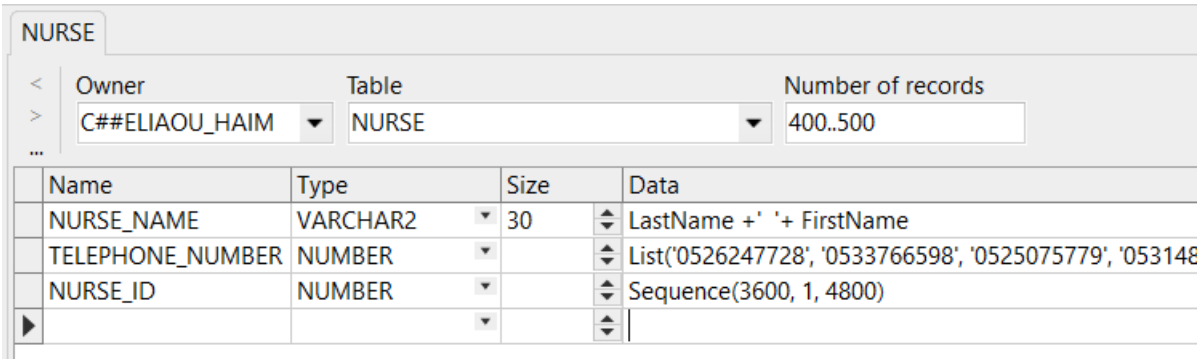
Name	Type	Size	Data
DOCTOR_NAME	VARCHAR2	30	LastName +' '+ FirstName
SPECIALTY	VARCHAR2	50	List('Cardiothoracic Surgery', 'General Surgery',
DOCTOR_ID	NUMBER		Sequence(2400, 1, 3600)

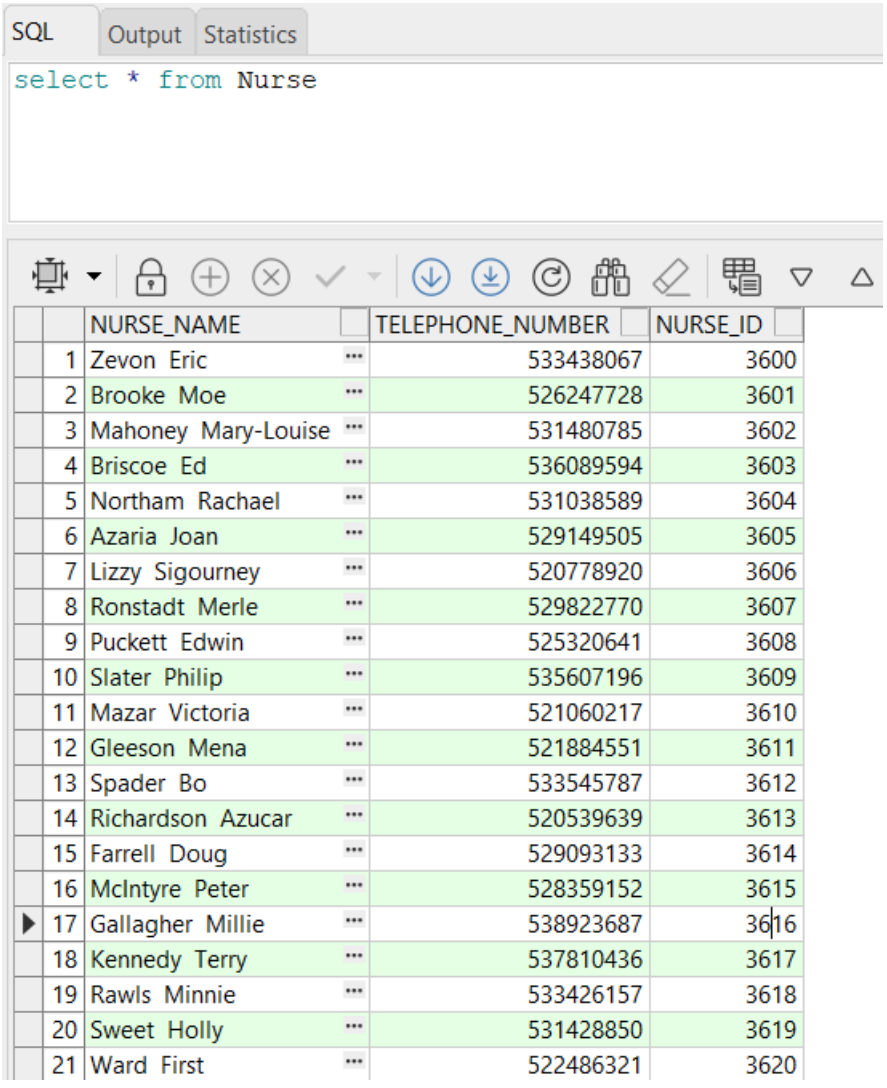
SQL | Output | Statistics

```
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:

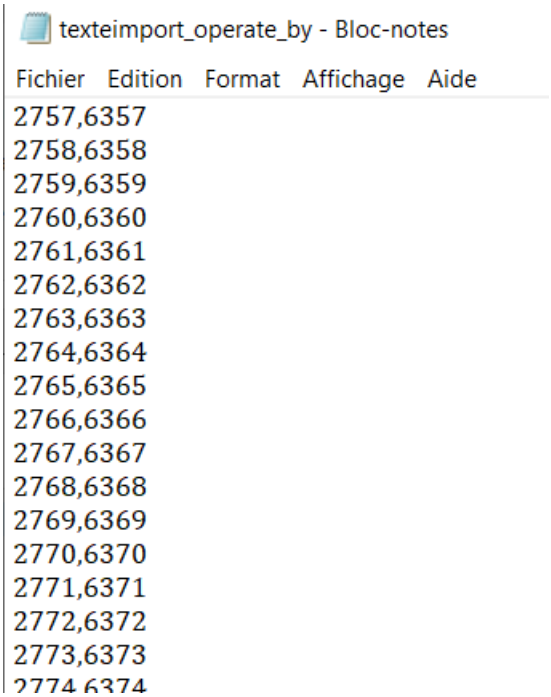


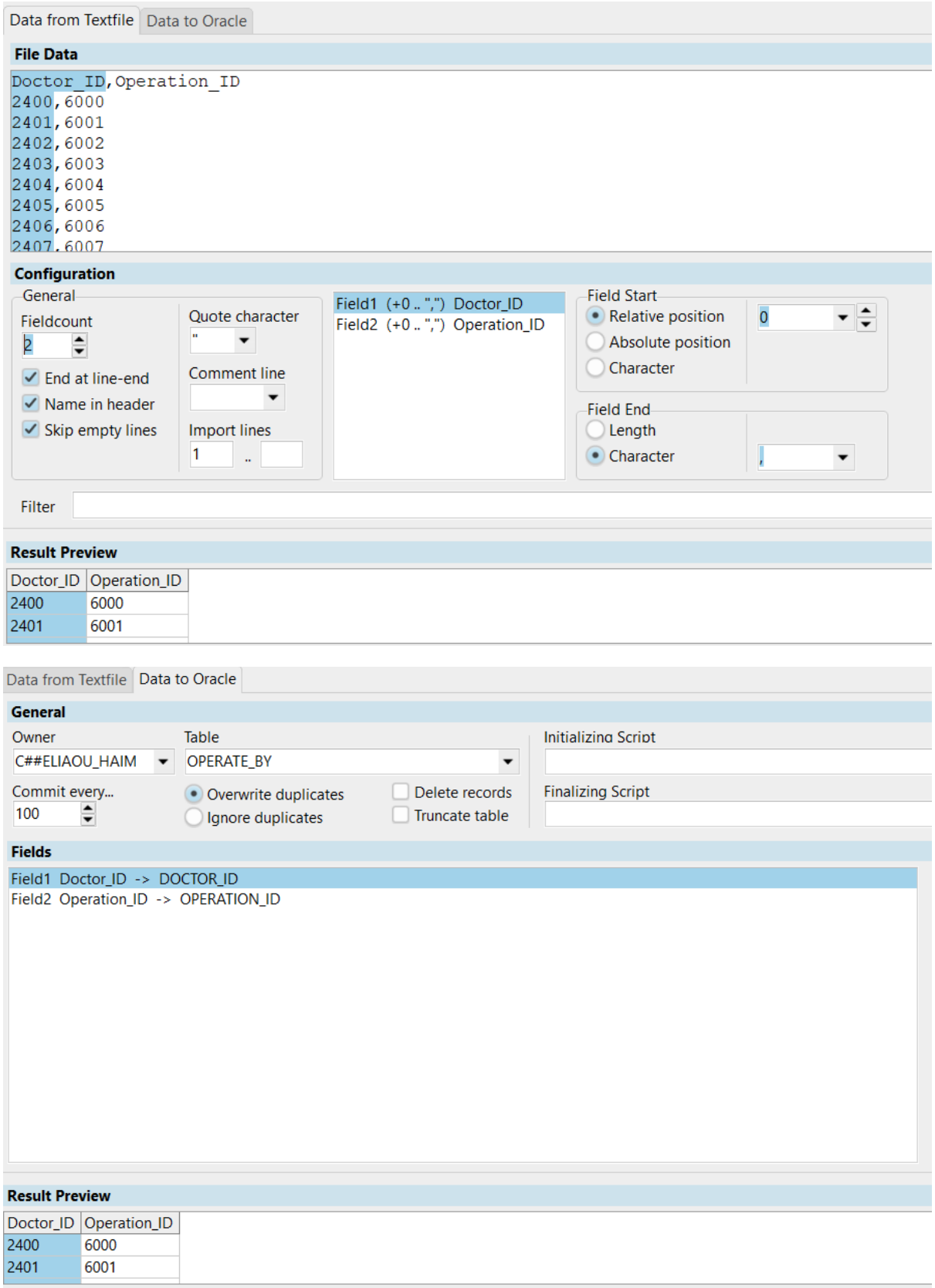


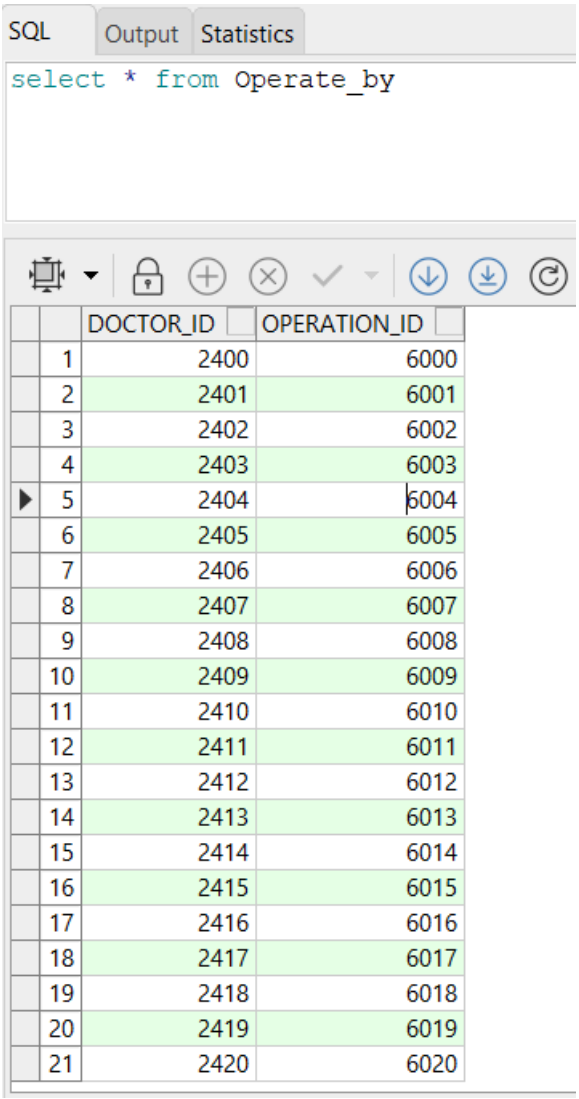
	NURSE_NAME	TELEPHONE_NUMBER	NURSE_ID
1	Zevon Eric	533438067	3600
2	Brooke Moe	526247728	3601
3	Mahoney Mary-Louise	531480785	3602
4	Briscoe Ed	536089594	3603
5	Northam Rachael	531038589	3604
6	Azaria Joan	529149505	3605
7	Lizzy Sigourney	520778920	3606
8	Ronstadt Merle	529822770	3607
9	Puckett Edwin	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:




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
Table
OPERATE_BY
Initializing Script
Commit every...
100
Overwrite duplicates
Ignore duplicates
Delete records
Truncate table
Finalizing Script
Fields
Field1 Doctor_ID -> DOCTOR_ID
Field2 Operation_ID -> OPERATION_ID
Result Preview
Doctor_ID
Operation_ID
2400
6000
2401
6001



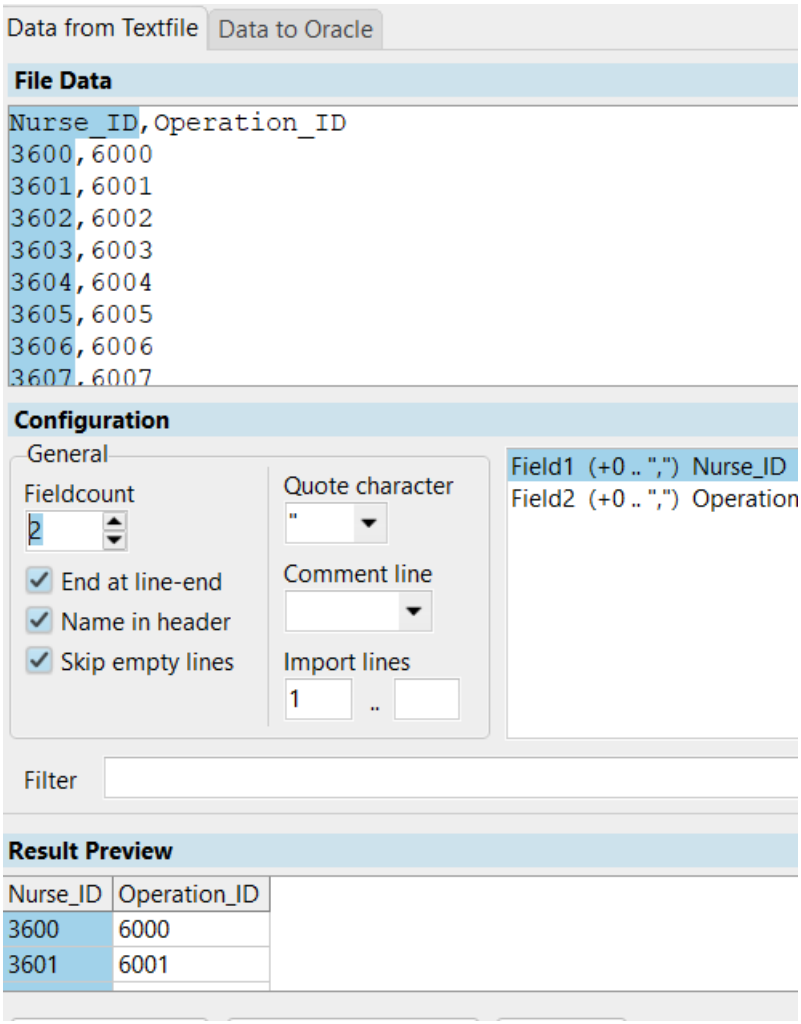
	DOCTOR_ID	OPERATION_ID
1	2400	6000
2	2401	6001
3	2402	6002
4	2403	6003
5	2404	6004
6	2405	6005
7	2406	6006
8	2407	6007
9	2408	6008
10	2409	6009
11	2410	6010
12	2411	6011
13	2412	6012
14	2413	6013
15	2414	6014
16	2415	6015
17	2416	6016
18	2417	6017
19	2418	6018
20	2419	6019
21	2420	6020

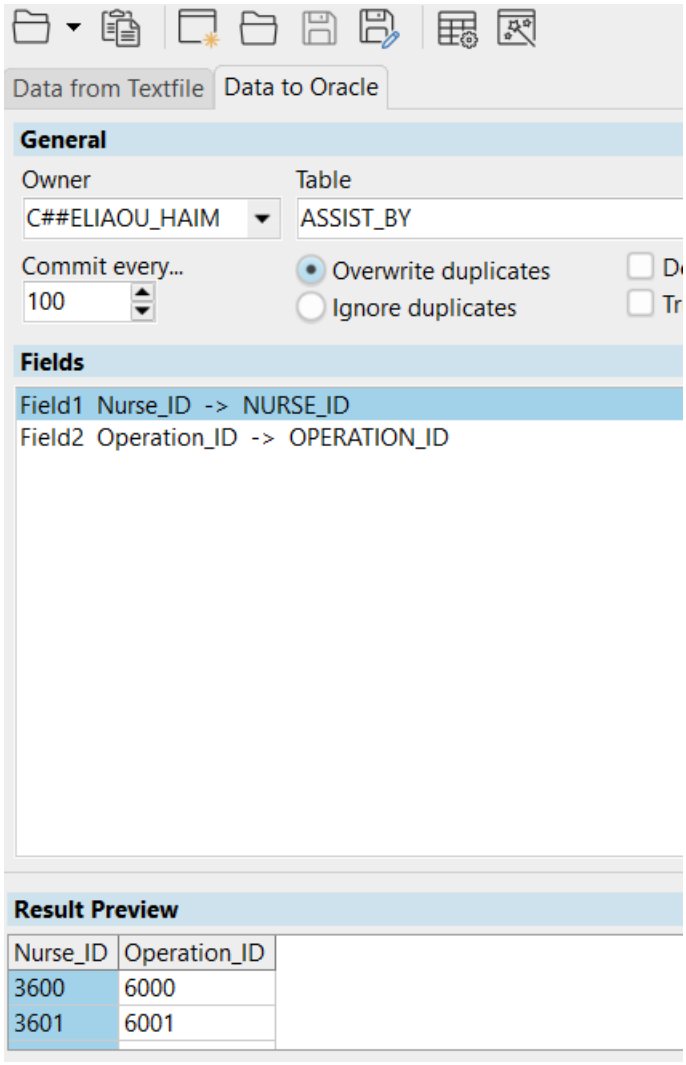
Inserting data into the Assit_by table:

texteimport_assist_by - Bloc-notes

Fichier Edition Format Affichage Aide

```
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
```




Data from Textfile
Data to Oracle
General
Owner
C##ELIAOU_HAIM
Table
ASSIST_BY
Commit every...
100
Overwrite duplicates
Ignore duplicates
Fields
Field1 Nurse_ID -> NURSE_ID
Field2 Operation_ID -> OPERATION_ID
Result Preview
Nurse_ID
Operation_ID
3600
6000
3601
6001

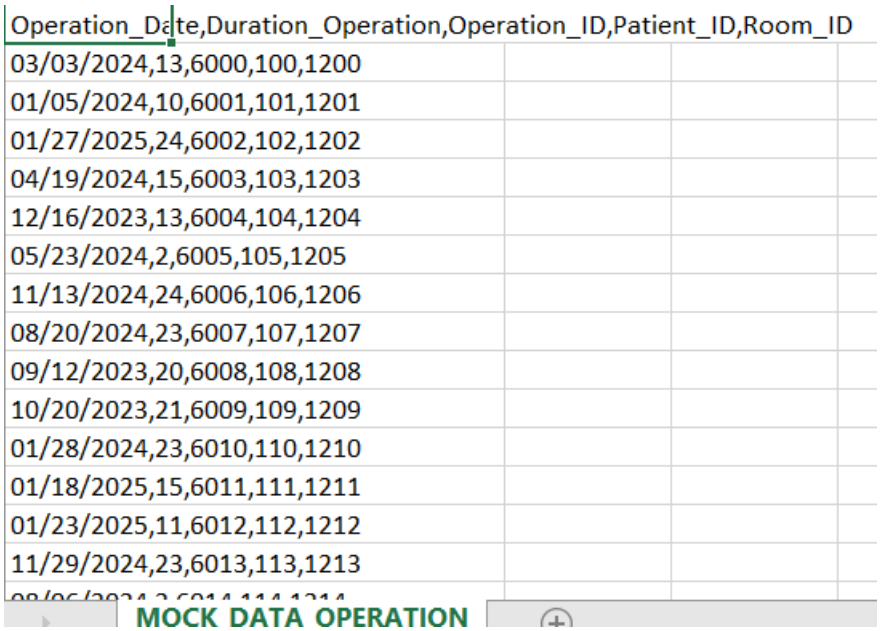
SQL | Output | Statistics

```
select * from Assist_by
```

	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

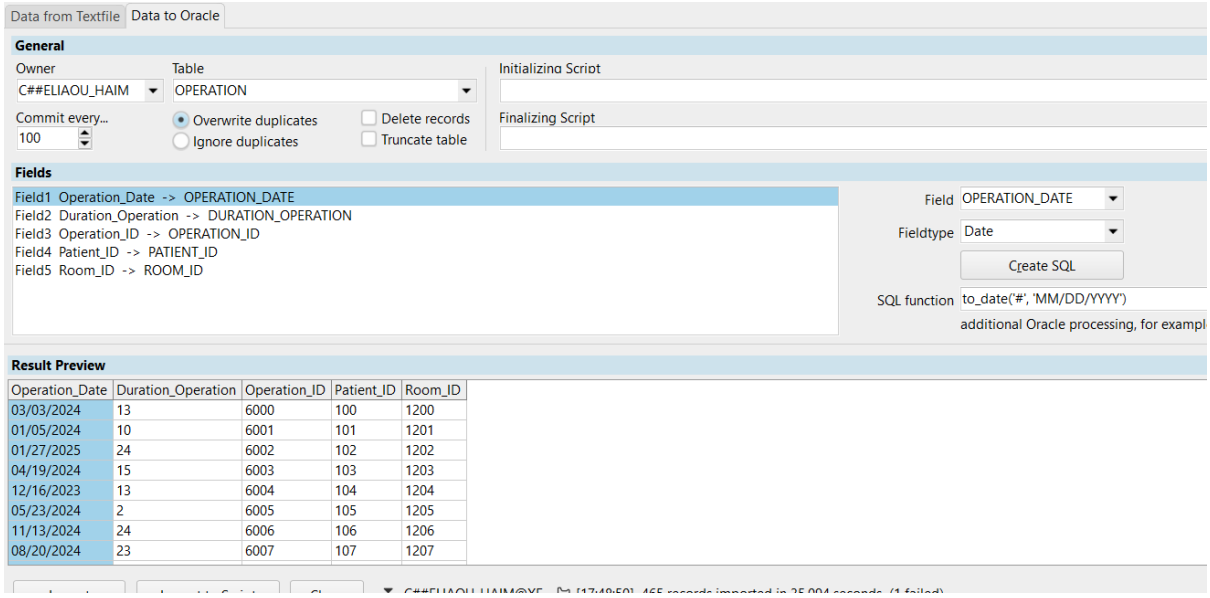
Entering data by EXCEL (mockaroo) :

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
12/06/2024	2	6014	114	1214
11/29/2024	23	6013	113	1213

MOCK DATA OPERATION



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: 5

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

Quote character: "

Comment line:

Import lines: 1 ..

Field1 (+0 .. ",") Operation_Date
Field2 (+0 .. ",") Duration_Operat
Field3 (+0 .. ",") Operation_ID
Field4 (+0 .. ",") Patient_ID
Field5 (+0 .. ",") Room_ID

Field Start: Relative position (0) / Absolute position / Character

Field End: Length / Character (,)

Filter

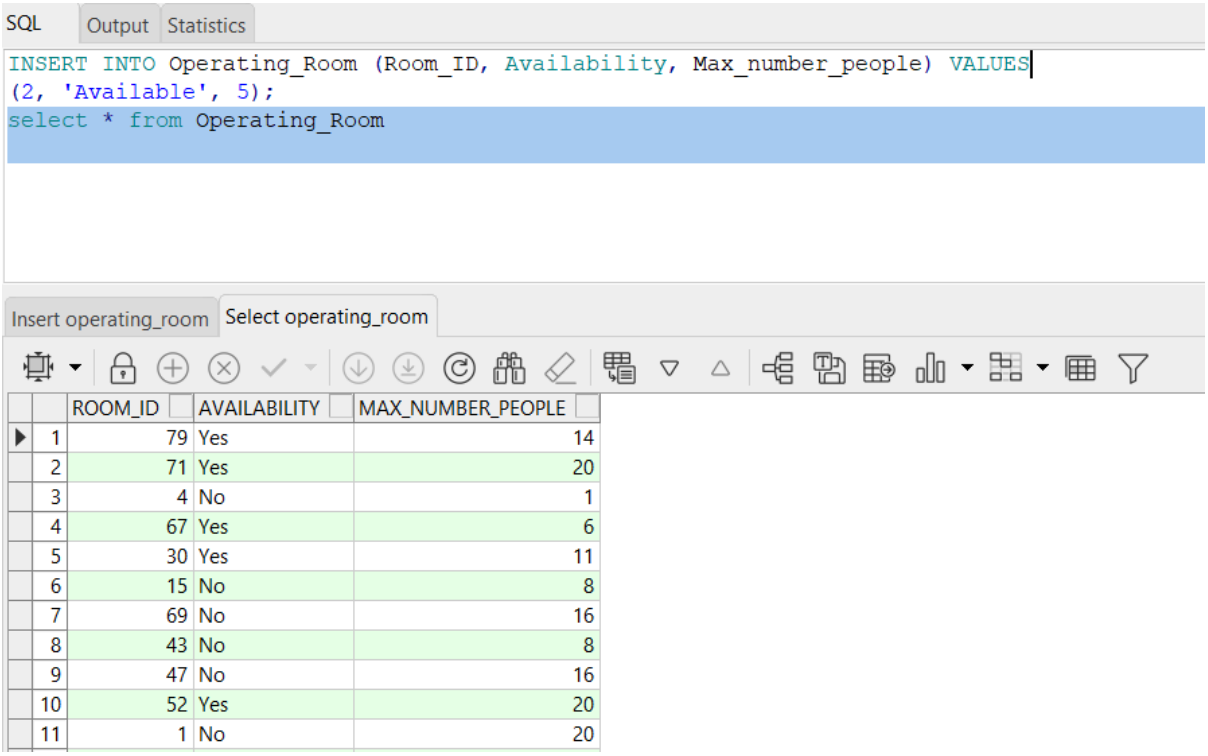
Result Preview

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:



	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:

שאילתות :

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	Evett Tzi	McGinley Davey	15/06/2023	9
3	Dalley Mindy	Lennix Toshiro	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 Equipment_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 Equipment_Status = 'maintenance'
WHERE Room_ID = 1415
  AND Equipment_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.

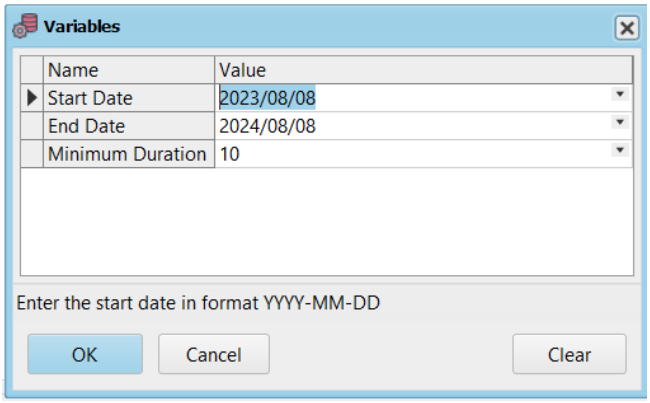
```
UPDATE Operating_Room
SET Availability = 'available'
WHERE Room_ID NOT IN (
    SELECT Room_ID
    FROM Operation
    WHERE Operation_Date > ADD_MONTHS(SYSDATE, -6)
);
```

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

שאילתות עם פרמטרים:

ParamsQueries.sql:

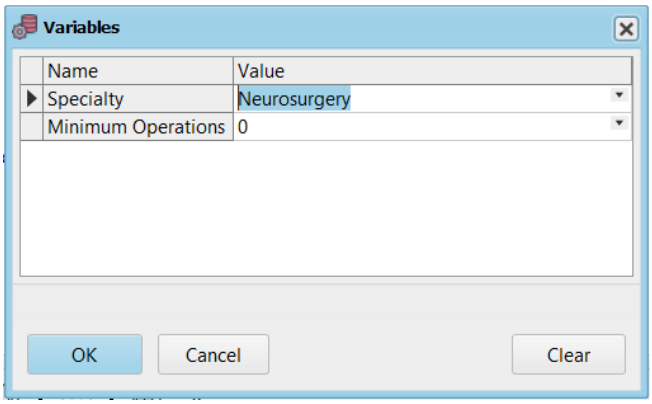
1. Operations between specific dates with a minimum duration




```
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;
```

	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

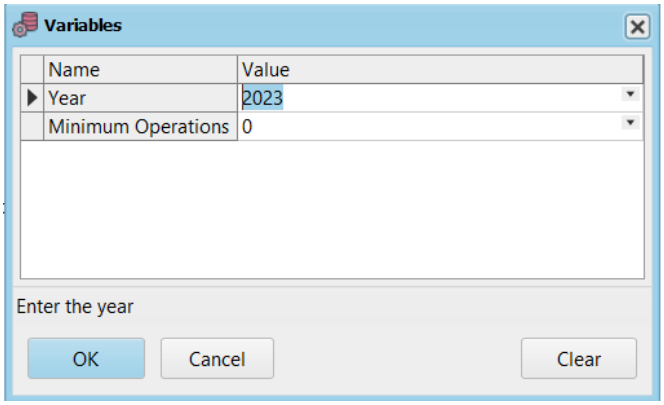
2. Doctors with a specific specialty who performed more than a certain number of operations.




```
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

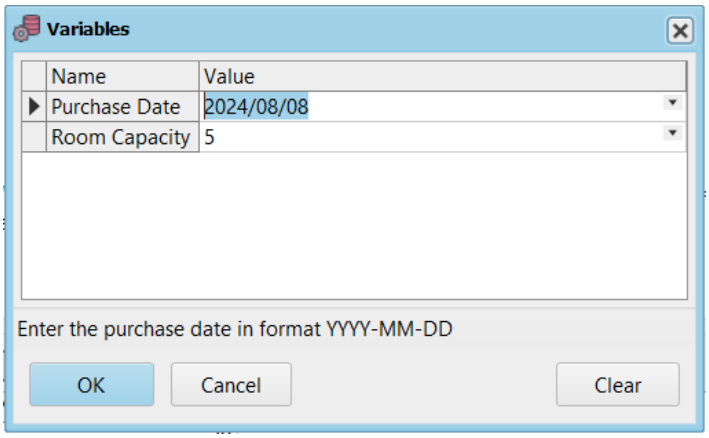
3. Nurses with a minimum number of operations assisted in a specific year.



```
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.




```
SELECT
    e.Equipement_ID,
    e.Equipment_Name,
    e.Equipment_Purchase_Date,
    r.Room_ID,
    r.Max_number_people
FROM
    Equipement 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;
```

	EQUIPEMENT_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:

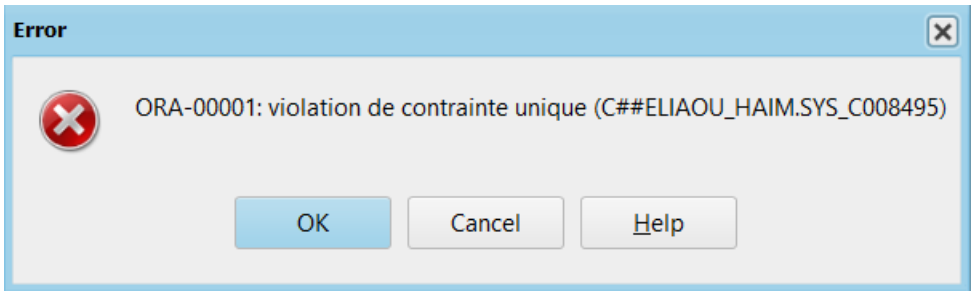
1. Adding CHECK constraints on `Operation`.

```
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;
```



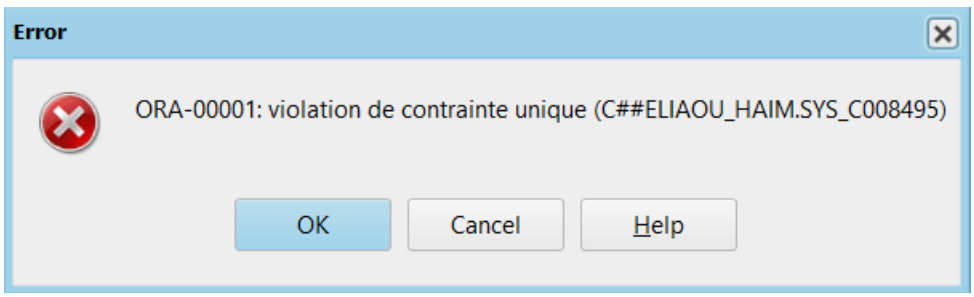
2. Adding UNIQUE constraint on `Doctor`

```
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;
```

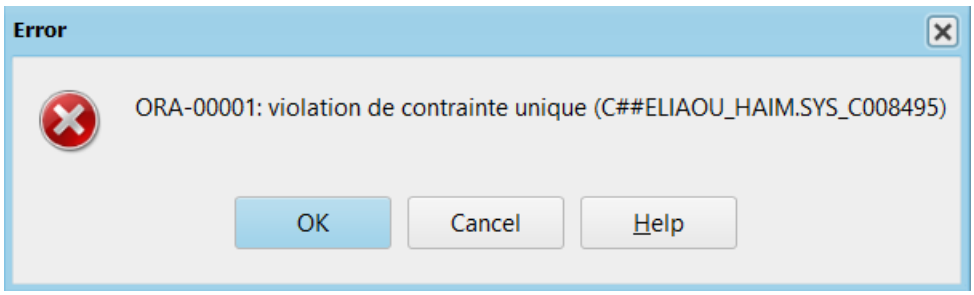


3. Adding DEFAULT constraint on `Operating_Room`

```
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:

For several years, the hospital has planned the renovation of certain operating rooms as well as the replacement of materials and equipment. As a result, several operating rooms and surgical equipment will be temporarily unavailable. It will therefore be necessary to redirect patients to available rooms and reassign equipment to ensure that each room has the necessary equipment.

The main program:

```
DECLARE
    RoomIDs SYS.ODCINUMBERLIST;
    EquipIDs SYS.ODCINUMBERLIST;
    MaintenanceStatus BOOLEAN;
BEGIN
    -- Simulate user input for room and equipment IDs
    RoomIDs := SYS.ODCINUMBERLIST(1212); -- room IDs to put in maintenance
    EquipIDs := SYS.ODCINUMBERLIST(4800); -- equipment IDs to put in maintenance

    -- Update maintenance status
    MaintenanceStatus := UpdateMaintenanceStatus(RoomIDs, EquipIDs);

    IF MaintenanceStatus THEN
        -- Redirect patients and staff
        RedirectOperationsAndStaff(RoomIDs, EquipIDs);
        DBMS_OUTPUT.PUT_LINE('Maintenance update and redirection completed successfully.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Failed to update maintenance status.');
    END IF;
END;
```

A function :

```
CREATE OR REPLACE FUNCTION UpdateMaintenanceStatus(
    RoomIDs IN SYS.ODCINUMBERLIST,
    EquipIDs IN SYS.ODCINUMBERLIST
) RETURN BOOLEAN IS
BEGIN
    -- Update rooms to maintenance status
    FOR i IN 1 .. RoomIDs.COUNT LOOP
        UPDATE Operating_Room
        SET Availability = 'Maintenance'
        WHERE Room_ID = RoomIDs(i);

        -- Display the updated room ID
        DBMS_OUTPUT.PUT_LINE('Room ID ' || RoomIDs(i) || ' set to Maintenance');
    END LOOP;

    -- Update equipment to maintenance status
    FOR i IN 1 .. EquipIDs.COUNT LOOP
        UPDATE Equipement
        SET Equipment_Status = 'Maintenance'
        WHERE Equipement_ID = EquipIDs(i);

        -- Display the updated equipment ID
        DBMS_OUTPUT.PUT_LINE('Equipment ID ' || EquipIDs(i) || ' set to Maintenance');
    END LOOP;

    RETURN TRUE;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Failed to update Update Maintenance Status: ' || SQLERRM);
        RETURN FALSE;
END UpdateMaintenanceStatus;
```

A procedure :

```sql
CREATE OR REPLACE PROCEDURE RedirectOperationsAndStaff(
    RoomIDs IN SYS.ODCINUMBERLIST,
    EquipIDs IN SYS.ODCINUMBERLIST
) IS
    CURSOR PatientsInMaintenanceRooms IS
        SELECT Patient_ID, Room_ID
        FROM Operation
        WHERE Room_ID IN (SELECT COLUMN_VALUE FROM TABLE(RoomIDs));

    PatientID Operation.Patient_ID%TYPE;
    OldRoomID Operation.Room_ID%TYPE;
    NewRoomID Operation.Room_ID%TYPE;

    OldEquipRoomID Equipement.Room_ID%TYPE;
    NewEquipID Equipement.Equipement_ID%TYPE;
    EquipmentName Equipement.Equipment_Name%TYPE;
BEGIN
    -- Redirect patients
    OPEN PatientsInMaintenanceRooms;
    LOOP
        FETCH PatientsInMaintenanceRooms INTO PatientID, OldRoomID;
        EXIT WHEN PatientsInMaintenanceRooms%NOTFOUND;

        BEGIN
            -- Find a new available room
            SELECT Room_ID
            INTO NewRoomID
            FROM Operating_Room
            WHERE Availability = 'Yes'
            AND ROWNUM = 1;

            UPDATE Operation
            SET Room_ID = NewRoomID
            WHERE Patient_ID = PatientID AND Room_ID = OldRoomID;

            -- Update the status of the new room to 'No'
            UPDATE Operating_Room
```

```
            SET Availability = 'No'
            WHERE Room_ID = NewRoomID;

            -- Display the updated patient details
            DBMS_OUTPUT.PUT_LINE('Patient ID ' || PatientID || ' moved from Room ' || OldRoomID || ' t
        EXCEPTION
            WHEN NO_DATA_FOUND THEN
                DBMS_OUTPUT.PUT_LINE('No available room found for patient ' || PatientID);
        END;
    END LOOP;
    CLOSE PatientsInMaintenanceRooms;

    -- Redirect equipment
    FOR i IN 1 .. EquipIDs.COUNT LOOP
        SELECT Room_ID, Equipment_Name
        INTO OldEquipRoomID, EquipmentName
        FROM Equipement
        WHERE Equipement_ID = EquipIDs(i);

        BEGIN
            -- Find a new available equipment with the same name
            SELECT Equipement_ID
            INTO NewEquipID
            FROM Equipement
            WHERE Equipment_Name = EquipmentName
            AND Equipment_Status = 'available'
            AND ROWNUM = 1;

            UPDATE Equipement
            SET Equipment_Status = 'In Use', Room_ID = OldEquipRoomID
            WHERE Equipement_ID = NewEquipID;

            -- Display the updated equipment details
            DBMS_OUTPUT.PUT_LINE('Equipment ID ' || EquipIDs(i) || ' replaced by Equipment ID ' || New
        EXCEPTION

            WHEN NO_DATA_FOUND THEN
                DBMS_OUTPUT.PUT_LINE('No available replacement found for Equipment ID ' || EquipIDs(i)
        END;
    END LOOP;
END RedirectOperationsAndStaff;
```

Running the program

The data before the test:

for example we want to put room 1212 and equipment 4800 into maintenance.

```
select * from Equipement
where equipment_name = 'Surgical sutures'
```

	EQUIPEMENT_ID	EQUIPMENT_PURCHASE_DATE	EQUIPMENT_NAME	EQUIPMENT_STATUS	ROOM_ID
1	4800	10/08/2024	Surgical sutures	available	1431
2	4875	06/10/2024	Surgical sutures	available	1546
3	4948	08/09/2024	Surgical sutures	not available	1463
4	5074	19/05/2024	Surgical sutures	available	1294
5	5152	18/05/2024	Surgical sutures	available	1503
6	5166	05/10/2024	Surgical sutures	available	1391

```
select * from Operating_Room
```

	ROOM_ID	AVAILABILITY	MAX_NUMBER_PEOPLE
1	1200	No	1
2	1201	No	20
3	1202	Maintenance	2
4	1203	Maintenance	11
5	1204	No	5
6	1205	Maintenance	7
7	1206	Yes	6
8	1207	Maintenance	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

We see that the room and the equipment are available.

	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	1206
4	19/04/2024	15	6003	103	1206
5	16/12/2023	13	6004	104	1204
6	23/05/2024	2	6005	105	1206
7	13/11/2024	24	6006	106	1206
8	20/08/2024	23	6007	107	1206
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

We also see that the patient 112 will have en operation in the room 1212.

The data after the test:

Test script | DBMS Output | Statistics | Profiler | Trace

Clear | Buffer size 10000 | Enabled

```
Room ID 1212 set to Maintenance
Equipment ID 4800 set to Maintenance
Patient ID 112 moved from Room 1212 to Room 1206
Equipment ID 4800 replaced by Equipment ID 4875 in Room 1431
Maintenance update and redirection completed successfully.
```

13	1212	Maintenance	9

The room 1212 is now in ‘maintenace’

```
select * from Equipement
where equipment_name ='Surgical sutures'
```

	EQUIPEMENT_ID	EQUIPMENT_PURCHASE_DATE	EQUIPMENT_NAME	EQUIPMENT_STATUS	ROOM_ID
1	4800	10/08/2024	Surgical sutures	Maintenance	1431
2	4875	06/10/2024	Surgical sutures	In Use	1431
3	4948	08/09/2024	Surgical sutures	not available	1463
4	5074	19/05/2024	Surgical sutures	available	1294
5	5152	18/05/2024	Surgical sutures	available	1503
6	5166	05/10/2024	Surgical sutures	available	1391

We see that the equipement 4800 changed to maintenance and an over equipment ’ Surgical sutures’ that was available have placed in the room 1431.

```
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	1206
4	19/04/2024	15	6003	103	1206
5	16/12/2023	13	6004	104	1204
6	23/05/2024	2	6005	105	1206
7	13/11/2024	24	6006	106	1206
8	20/08/2024	23	6007	107	1206
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	1206

We see now that the room for the operation of the patient 112 have changed to 1206 that is the first room with the avaibility yes. (and the room 1212 have no operation now).

	ROOM_ID	AVAILABILITY
1	1200	No
2	1201	No
3	1202	Maintenance
4	1203	Maintenance
5	1204	No
6	1205	Maintenance
7	1206	No

Now the status of the room 1206 that was Yes is now No.

תוכנית ב:

Program description:

We have identified a doctor whose operations have all failed and who has an unusually high number of deceased patients. This doctor now represents too high a cost for our hospital in terms of litigation. Therefore, we have decided to remove this doctor from our list of doctors and reassign all his patients to other available doctors specialized in the same discipline, while ensuring that these doctors were not having other operations.

The main program:

```
BEGIN
    -- ID of the doctor to remove and reassign patients
    ReassignPatientsAndRemoveDoctor(2400);

    DBMS_OUTPUT.PUT_LINE('Test completed.');
END;
```

A function:

```
CREATE OR REPLACE PROCEDURE ReassignPatientsAndRemoveDoctor(
    p_Doctor_ID IN Doctor.Doctor_ID%TYPE
) IS
    TYPE PatientRec IS RECORD (
        Patient_ID Patient.Patient_ID%TYPE,
        Operation_ID Operation.Operation_ID%TYPE,
        Operation_Date Operation.Operation_Date%TYPE,
        Old_Room_ID Operation.Room_ID%TYPE,
        Specialty Doctor.Specialty%TYPE
    );

    CURSOR PatientCursor IS
        SELECT o.Patient_ID, o.Operation_ID, o.Operation_Date, o.Room_ID, d.Specialty
        FROM Operation o
        JOIN Operate_by ob ON o.Operation_ID = ob.Operation_ID
        JOIN Doctor d ON ob.Doctor_ID = d.Doctor_ID
        WHERE ob.Doctor_ID = p_Doctor_ID;

    PatientRecord PatientRec;
    Success BOOLEAN;
BEGIN
    OPEN PatientCursor;
    LOOP
        FETCH PatientCursor INTO PatientRecord;
        EXIT WHEN PatientCursor%NOTFOUND;

        -- Call function to reassign patient
        Success := ReassignPatient(
            p_Operation_ID => PatientRecord.Operation_ID,
            p_Specialty => PatientRecord.Specialty,
            p_Operation_Date => PatientRecord.Operation_Date
        );

        IF Success THEN
            DBMS_OUTPUT.PUT_LINE('Patient ID ' || PatientRecord.Patient_ID || ' re-assigned successful
        ELSE
            DBMS_OUTPUT.PUT_LINE('Failed to re-assign Patient ID ' || PatientRecord.Patient_ID);
        END IF;
    END LOOP;
    CLOSE PatientCursor;

    -- Delete the doctor from the list of doctors
    DELETE FROM Doctor
    WHERE Doctor_ID = p_Doctor_ID;

    DBMS_OUTPUT.PUT_LINE('Doctor ID ' || p_Doctor_ID || ' has been removed from the list of doctors.'

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error in ReassignPatientsAndRemoveDoctor: ' || SQLERRM);
END ReassignPatientsAndRemoveDoctor;
```

A procedure:

```sql
CREATE OR REPLACE FUNCTION ReassignPatient(
    p_Operation_ID IN Operation.Operation_ID%TYPE,
    p_Specialty IN Doctor.Specialty%TYPE,
    p_Operation_Date IN Operation.Operation_Date%TYPE
) RETURN BOOLEAN IS
    New_Room_ID Operating_Room.Room_ID%TYPE;
    New_Doctor_ID Doctor.Doctor_ID%TYPE;
BEGIN
    -- Find a new available room
    SELECT Room_ID
    INTO New_Room_ID
    FROM Operating_Room
    WHERE Availability = 'Yes'
    AND ROWNUM = 1;

    -- Find a new available doctor with the same specialty and available on the operation date
    SELECT Doctor_ID
    INTO New_Doctor_ID
    FROM Doctor d
    WHERE d.Specialty = p_Specialty
    AND NOT EXISTS (
        SELECT 1
        FROM Operate_by ob
        JOIN Operation o ON ob.Operation_ID = o.Operation_ID
        WHERE ob.Doctor_ID = d.Doctor_ID
        AND o.Operation_Date = p_Operation_Date
    )
    AND ROWNUM = 1;

    -- Update the operation with the new room and doctor
    UPDATE Operation
    SET Room_ID = New_Room_ID
    WHERE Operation_ID = p_Operation_ID;

    UPDATE Operate_by
    SET Doctor_ID = New_Doctor_ID
    WHERE Operation_ID = p_Operation_ID;
```

```sql
    -- Update the operation with the new room and doctor
    UPDATE Operation
    SET Room_ID = New_Room_ID
    WHERE Operation_ID = p_Operation_ID;

    UPDATE Operate_by
    SET Doctor_ID = New_Doctor_ID
    WHERE Operation_ID = p_Operation_ID;

    -- Update the status of the new room to 'No'
    UPDATE Operating_Room
    SET Availability = 'No'
    WHERE Room_ID = New_Room_ID;

    -- Display the updated patient and operation details
    DBMS_OUTPUT.PUT_LINE('Operation ID ' || p_Operation_ID || ' re-assigned to Doctor ID ' || New_Doct

    RETURN TRUE;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No available room or doctor found for Operation ID ' || p_Operation_ID);
        RETURN FALSE;
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error in ReassignPatient: ' || SQLERRM);
        RETURN FALSE;
END ReassignPatient;
```

Running the program

The data before the test:

For exemple supose that we want to delete the doctor with id 2400

```
select * from doctor
where specialty = 'General Surgery'
```

	DOCTOR_NAME	SPECIALTY	DOCTOR_ID
1	Laurie Jimmy	General Surgery	2400
2	Rizzo Alex	General Surgery	2448
3	McGill Garland	General Surgery	2512
4	Voight Viggo	General Surgery	2604
5	Ball Jodie	General Surgery	2608
6	Thurman Gabriel	General Surgery	2720
7	LaMond Salma	General Surgery	2728
8	Bracco Josh	General Surgery	2731

The doctor with id 2400 have the operation with id 6000

```
select * from operate_by
```

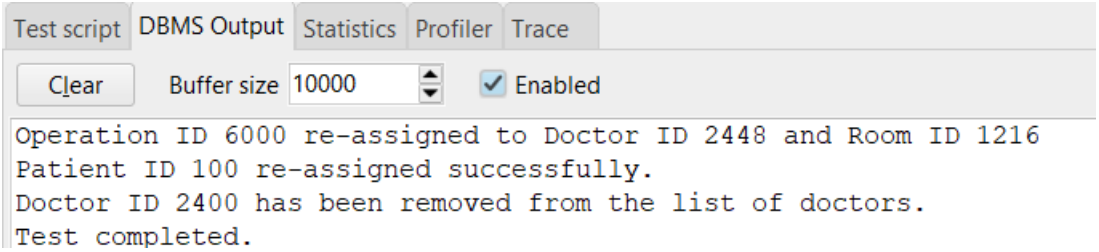
	DOCTOR_ID	OPERATION_ID
1	2400	6000

The patient with id 100 have the operation 6000 in the room id 1200

```
select * from operation
```

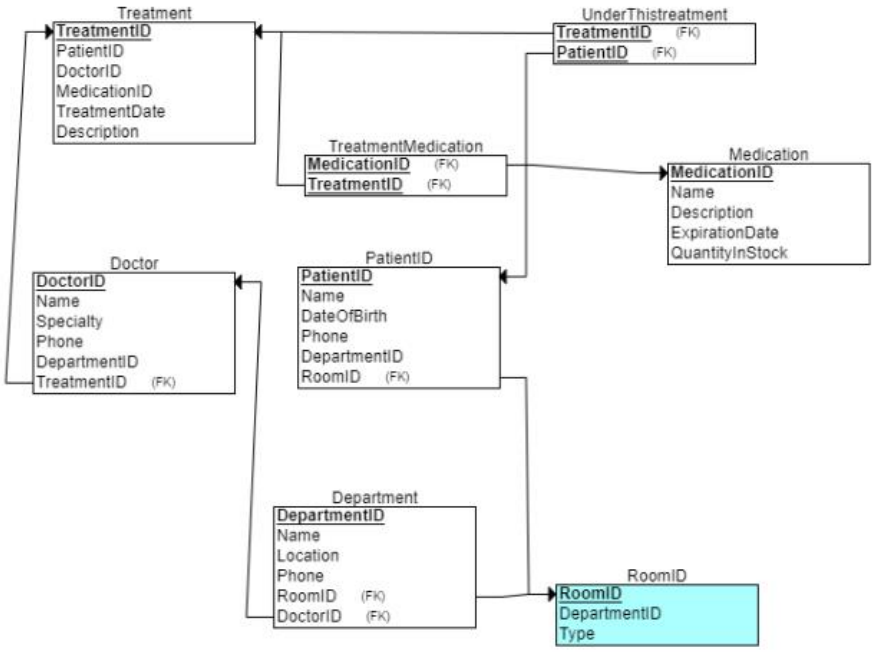
	OPERATION_DATE	DURATION_OPERATION	OPERATION_ID	PATIENT_ID	ROOM_ID
1	03/03/2024	13	6000	100	1200

The data after the test:

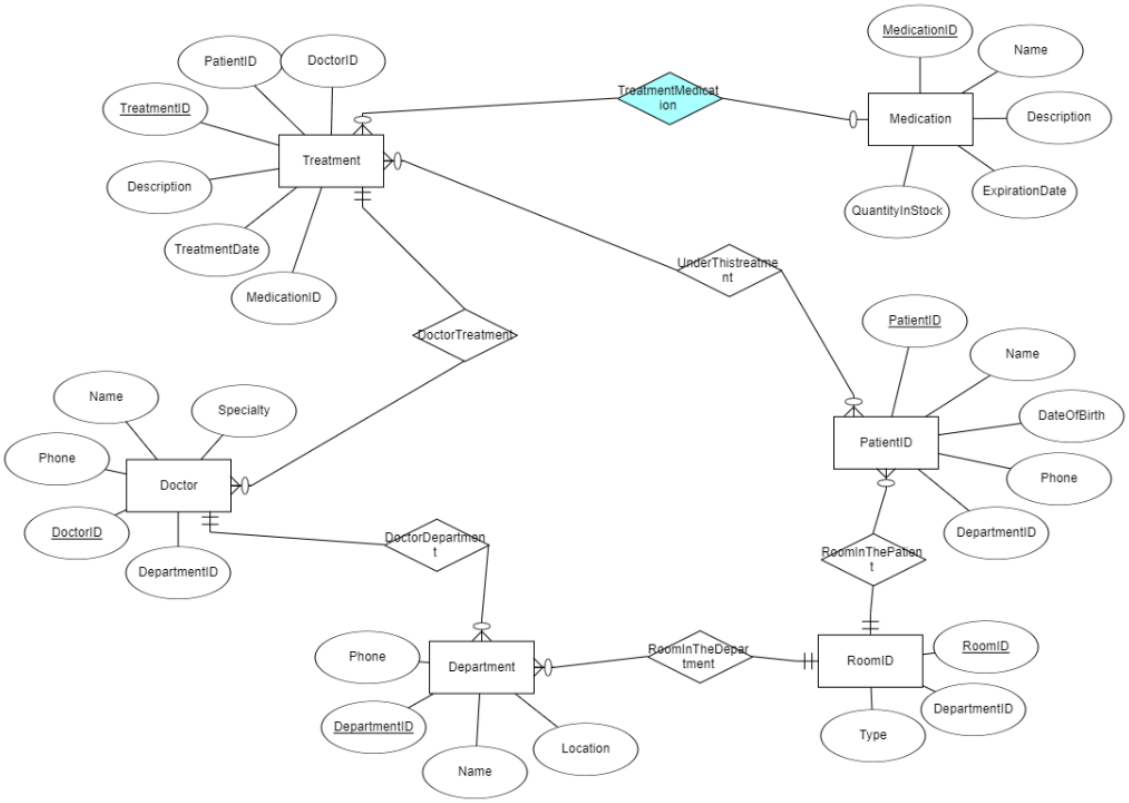


חלק 3:

האגף החדש - DSD :



האגף החדש - ERD :



המערכת המשותפת - ERD :

the entities included in the system

1. Department:

• DepartmentID: unique identifier for the department.

• Name: the name of the department (for example, cardiology, oncology).

• Location: the location of the department within the hospital.

• Phone: phone number of the department.

• RoomID (FK) unique identifier for the room.

• DoctorID (FK) unique identifier for a doctor.

2. Operating Room:

• RoomID: unique identifier for the room.

• Type: the type of room (for example, intensive care, hospitalization).

• DepartmentID: ID of the department to which the room belongs.

• Availability - Indicates if the room is available.

• Max_number_people - indicates the maximum number of people that the room can accommodate

3. Doctor:

• DoctorID: unique identifier for a doctor.

• Name: The doctor's name.

• Specialty: the doctor's specialty (for example, cardiology, neurology).

• Phone: phone number of the doctor.

• DepartmentID: ID of the department to which the doctor is associated.

• TreatmentID: (FK) unique identifier for the treatment.

4. Patient:

• PatientID: unique identifier for the patient.

• Name: the patient's name.

• DateOfBirth: The patient's date of birth.

• Phone: phone number of the patient.

• DepartmentID: the identifier of the department to which the patient is associated.

• Sexe - The sexe of the patient

• Illness - Brief description of the subject of the operation.

• RoomID (FK) unique identifier for the room.

5. Medication:

• MedicationID: (K) unique identifier for the medication.

• Name: the name of the medicine.

• Description: Description of the medicine.

• ExpirationDate: expiration date of the drug.

• QuantityInStock: the quantity of the drug in stock.

6. Treatment:

• TreatmentID: (K) unique identifier for the treatment.

• PatientID: the ID of the patient receiving the treatment.

• DoctorID: ID of the attending physician.

• MedicationID: ID of the medication given in the treatment.

• TreatmentDate: the treatment date.

• Description: description of the treatment.

7. UnderThisTreatment:

• TreatmentID: (FK) unique identifier for the treatment.

• PatientID: (FK) the ID of the patient receiving the treatment.

8. TreatmentMedication:

• MedicationID: (FK) unique identifier for the medication.

• TreatmentID: (FK) unique identifier for the treatment.

9.Nurse :

• Nurse_ID (PK) - Nurse's identification number

• Nurse_Name - The nurse name

• Telephone_number - The telephone number of the nurse

10. Assist_by:

• Nurse_ID (FK) - Nurse's identification number

• Operation_ID (FK) - Operation's identification number

11. Operate_by:

• Doctor_ID (FK) - Doctor's identification number

• Operation_ID (FK) - Operation's identification number

12. Equipement :

• 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.

• RoomID (O)(FK) unique identifier for the room.

13. Operation ：

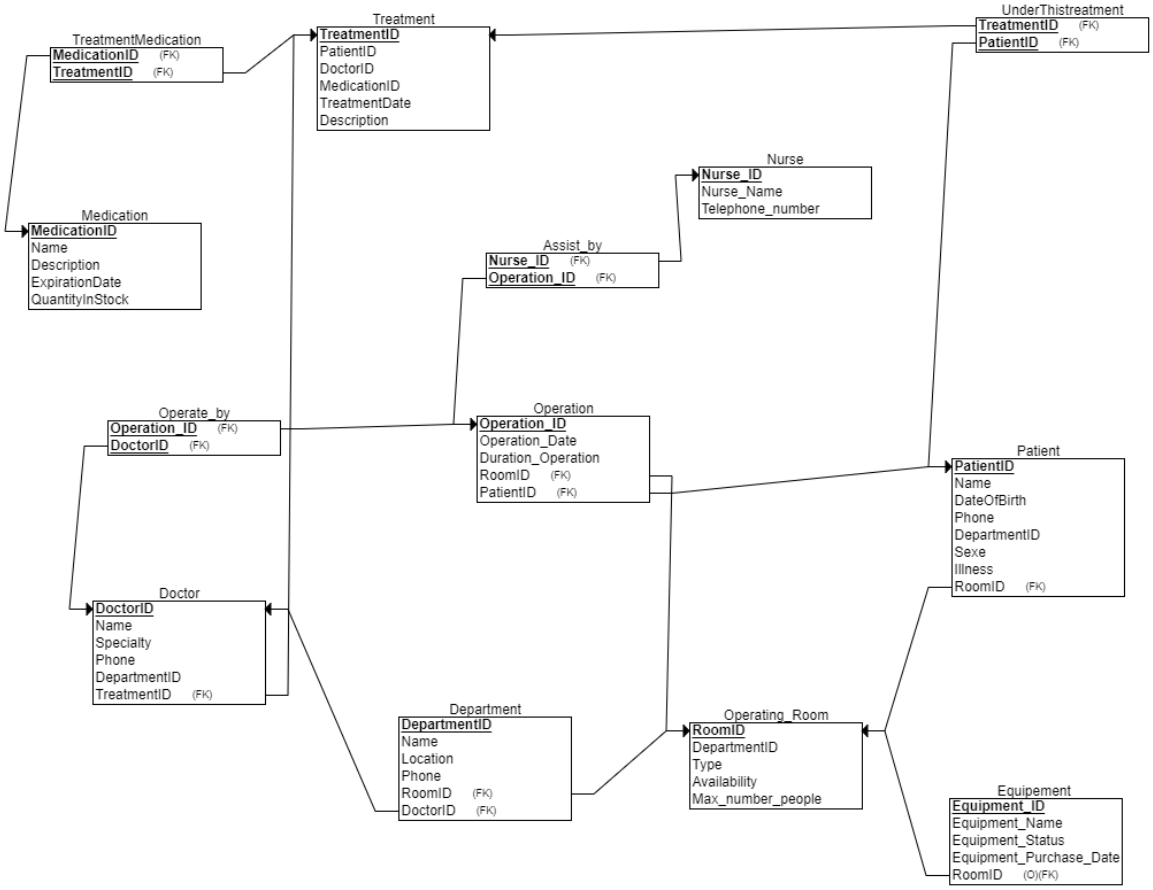
- Operation_ID (PK) - Operation’s identification number
- Operation_Date - The date of the operation
- Duration_Operation - The time that takes the operation.
- RoomID (FK) unique identifier for the room.
- PatientID (FK) unique identifier for the patient.

Integrated ERD:

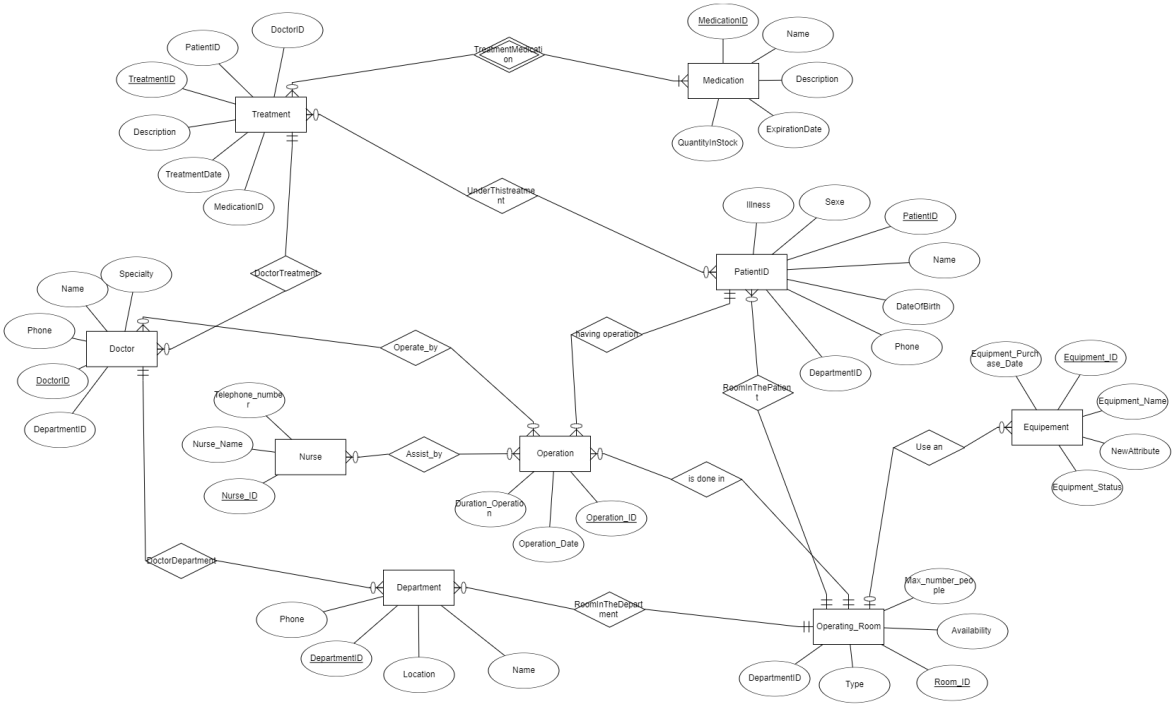
First, we added the new tables. Next, we added the new foreign keys to the existing tables. After that, we performed an import of backup3 for the team and recovered the data. Several problems emerged: missing entries in some new tables, a lack of common keys in many tables, differing attribute names, and inconsistent IDs.

To resolve all these problems, we had to execute numerous SQL commands and modify certain parameters, as well as change table names and attributes.

המערכת המשותפת - DSD :



המערכת המשותפת - ERD :



Here is someone (you can find the complete details on GitHub in the `integrate.sql` file.):

```sql
ALTER TABLE Patient
ADD DateOfBirth DATE,
ADD Phone INT,
ADD DepartmentID INT,
ADD RoomID INT;

ALTER TABLE Patient
ADD FOREIGN KEY (RoomID) REFERENCES Operating_Room(RoomID);
```

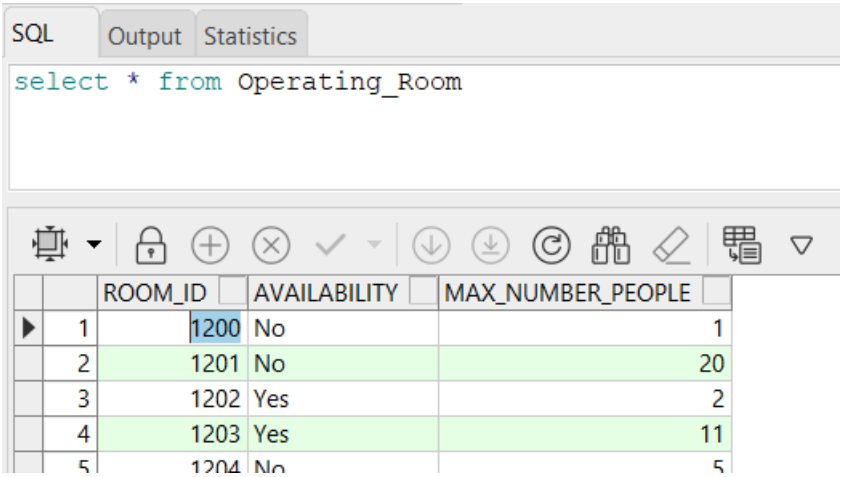
```sql
DELETE FROM PATIENT
WHERE ROOMID IS NOT NULL
AND ROOMID NOT IN (SELECT ROOMID FROM operating_room);
```

```sql
--- To add DEPARTMENTID to doctor starting with 1 ---

MERGE INTO doctor p
USING (
    SELECT ROWID, ROW_NUMBER() OVER (ORDER BY DEPARTMENTID) + 0 AS new_patientid
    FROM doctor
) u
ON (p.ROWID = u.ROWID)
WHEN MATCHED THEN
UPDATE SET p.DEPARTMENTID = u.new_patientid;
```

...(integrate.sql)

Here is some table befor and after their integration:



	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

After:

```
select * from operating_room
```

	ROOMID	AVAILABILITY	MAX_NUMBER_PEOPLE	DEPARTMENTID	TYPE
1	1200	No	1	1	NICU
2	1204	No	5	5	Recovery
3	1205	Maintenance	7	6	Maternity
4	1207	Maintenance	5	8	PICU
5	1209	No	16	10	Recovery
6	1211	No	12	12	Isolation
7	1213	Maintenance	15	14	PICU
8	1214	No	9	15	Surgery
9	1217	No	5	18	PICU
10	1219	Yes	4	20	CCU

Befor:

SQL Output Statistics

```
select * from patient
```

	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

After:

```
select * from patient
```

	SEXE	ILLNESS	DATEOFBIRTH	PHONE	ROOMID	PATIENTID	NAME
1	Man	cholecystectomy	26/04/1967	3494612546	2162	1062	Daniel Anderson
2	Man	appendicitis	14/06/1959	2100980687	1201	101	Robert Williams
3	Woman	inguinal hernia	07/06/1954	3531943276	1203	103	Robert Martinez
4	Woman	appendicitis	07/02/1952	4000819784	1206	106	Ava Perez
5	Man	adrenalectomy	24/04/1971	6974713047	1208	108	Isaac Jones
6	Man	gallstones	19/04/1961	7627616702	1209	109	Sophia Lewis

All the tables can be found on the github for more details.

שאילתות 1 (מנקודת המבט שלנו):

Returns the name, patientid, doctor.name and the description of all the patients who received treatment that was given by the doctor.Name which contains the word "Eli".

```
---  Returns the name, patientid, doctor.name and the description of all the patients who

CREATE OR REPLACE VIEW PatientsTreatedByDoctorEli AS
SELECT p.PATIENTID, p.NAME AS PATIENT_NAME, d.NAME AS DOCTOR_NAME, m.DESCRIPTION
FROM PATIENT p
JOIN TREATMENT t ON p.PATIENTID = t.PATIENTID
JOIN DOCTOR d ON t.DOCTORID = d.DOCTORID
JOIN MEDICATION_TREATMENT mt ON t.TREATMENTID = mt.TREATMENTID
JOIN MEDICATION m ON mt.MEDICATIONID = m.MEDICATIONID
WHERE d.NAME LIKE '%Eli%';

-- Return patients treated by a doctor whose name contains "Eli"
SELECT PATIENTID, PATIENT_NAME, DOCTOR_NAME, DESCRIPTION
FROM PatientsTreatedByDoctorEli;
```

Create patientstreatedbydoctoreli | Select patientstreatedbydoctoreli

	PATIENTID	PATIENT_NAME	DOCTOR_NAME	DESCRIPTION
1	307	Luke Perez	Channing Eliza	Replacement of Up Tooth, All, with Autol Sub, Open Approach
2	398	Aria Evans	Sayer Elizabeth	Drainage of Transverse Colon with Drain Dev, Via Opening
3	455	Amelia White	Koteas Elizabeth	Excision of R Temporomandib Jt, Perc Endo Approach, Diagn
4	546	Hannah Green	Elita Gerding	Replace of Pancreat Duct with Synth Sub, Perc Endo Approach

שאילתות 2 (מנקודת המבט שלנו):

This query returns patients for whom the TREATMENTS DATE is before or equal to OPERATION_DATE.

```
--  This query returns patients for whom the TREATMENTS DATE is

CREATE OR REPLACE VIEW TreatmentsBeforeOrEqualOperation AS
SELECT o.PATIENTID,
       o.OPERATION_DATE,
       t.TREATMENTDATE
FROM PATIENT p
JOIN OPERATION o ON p.PATIENTID = o.PATIENTID
JOIN TREATMENT t ON p.PATIENTID = t.PATIENTID
WHERE t.TREATMENTDATE <= o.OPERATION_DATE;

-- Return patients with a treatment date before or equal to the
SELECT PATIENTID, OPERATION_DATE, TREATMENTDATE
FROM TreatmentsBeforeOrEqualOperation;
```

Create treatmentsbeforeorequaloperation | Select treatmentsbeforeorequaloperation

	PATIENTID	OPERATION_DATE	TREATMENTDATE
1	100	03/03/2024	21/04/2009
2	101	05/01/2024	17/12/2016
3	102	27/01/2025	06/10/2002
4	103	19/04/2024	03/02/2022
5	104	16/12/2023	11/05/2022
6	105	23/05/2024	12/11/2016
7	106	13/11/2024	05/10/2023

(שאילתות 3 (מנקודת המבט השנייה):

This query allows you to find all the patients for whom you have had an operation but not yet received treatment.

```
--  This query allows you to find all the patients for whom

CREATE OR REPLACE VIEW PatientsWithoutTreatment AS
SELECT o.OPERATION_ID, o.PATIENTID
FROM OPERATION o
LEFT JOIN TREATMENT t ON o.PATIENTID = t.PATIENTID
WHERE t.PATIENTID IS NULL;

-- Find all patients without treatment
SELECT OPERATION_ID, PATIENTID
FROM PatientsWithoutTreatment;
```

Create patientswithouttreatment | Select patientswithouttreatment

	OPERATION_ID	PATIENTID
1	6024	124
2	6131	231
3	6262	362
4	6402	502

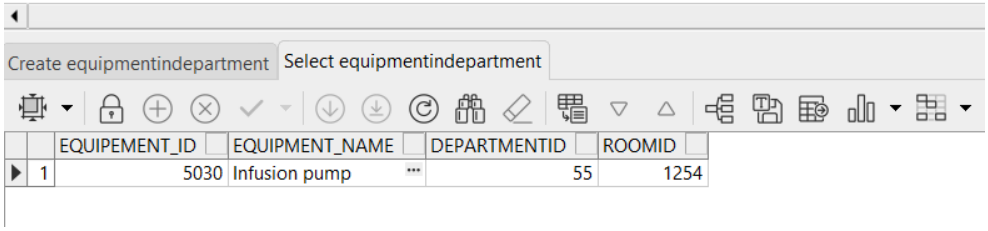
שאילתות 4 (מנקודת המבט השנייה):

This query allows you to display the equipment used in a department and in which room it is used. (for example we keep the departement “55”).

```
---  This query allows you to display the equipment used in a department and i

CREATE OR REPLACE VIEW EquipmentInDepartment AS
SELECT e.EQUIPEMENT_ID, e.EQUIPMENT_NAME, oroom.DEPARTMENTID, oroom.ROOMID
FROM EQUIPEMENT e
JOIN OPERATING_ROOM oroom ON e.ROOMID = oroom.ROOMID
WHERE oroom.DEPARTMENTID = 55;

-- View equipment used in a specific department
SELECT EQUIPEMENT_ID, EQUIPMENT_NAME, DEPARTMENTID, ROOMID
FROM EquipmentInDepartment;
```



	EQUIPEMENT_ID	EQUIPMENT_NAME	DEPARTMENTID	ROOMID
1	5030	Infusion pump	55	1254

THE END!!