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

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.

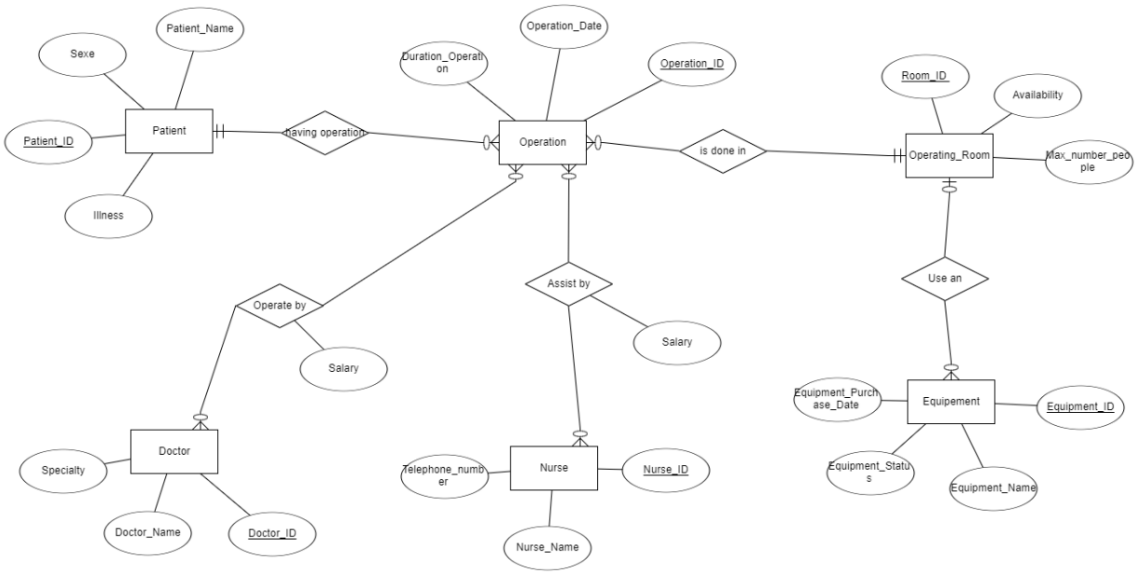
An operation takes place in a single room. (M: 1)
A room can accommodate several operations.

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

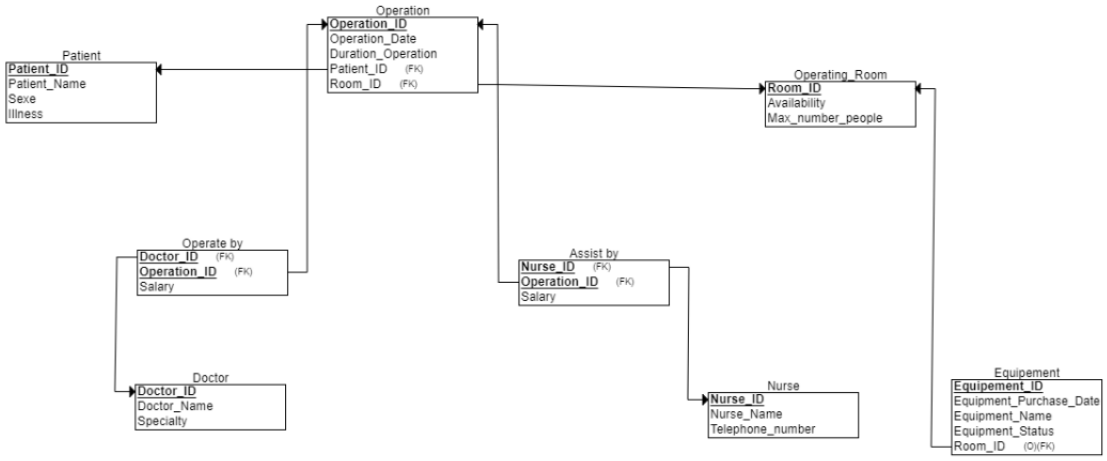
One piece of equipment can be used by a maximum of one operations room. (1: N)
An operating room may require several pieces of equipment.

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

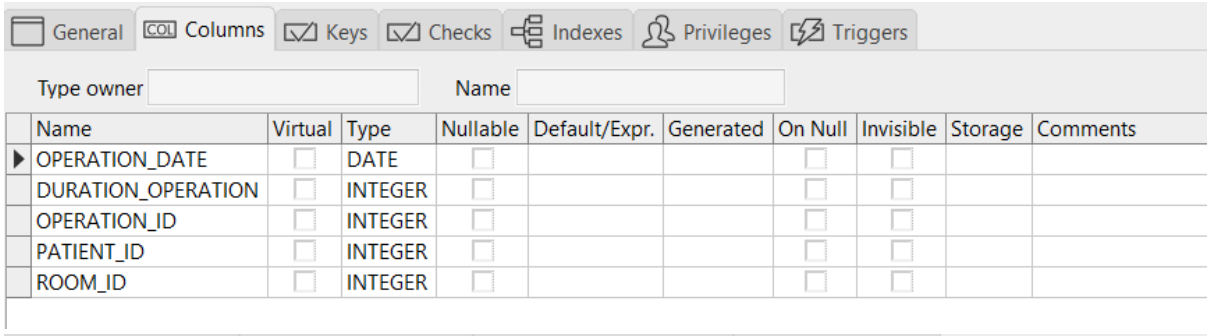
Name	Type	Columns	Enabled	Referencing table	Referencing columns	On Delete	Deferrable	Deferred	Validated	Last change
SYS_C008318	Primary	PATIENT_ID	☑				☐	☐	☑	27/05/2024 15:14:36

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

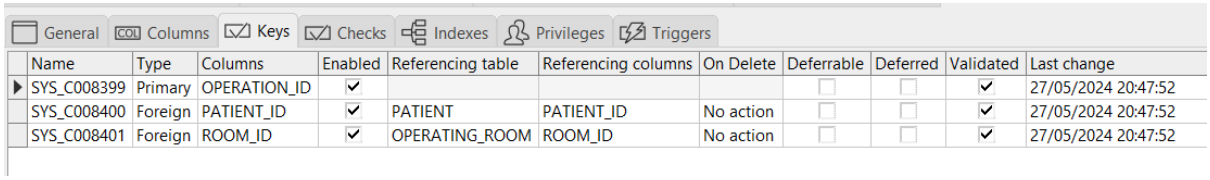
Type owner Name

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:



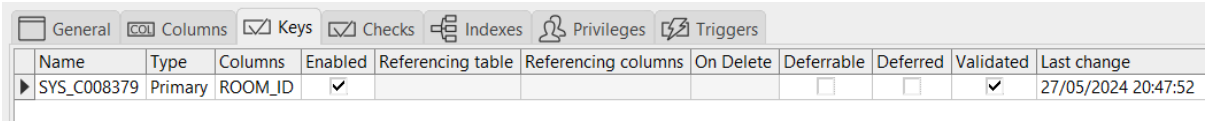
Name	Virtual	Type	Nullable	Default/Expr.	Generated	On Null	Invisible	Storage	Comments
OPERATION_DATE		DATE							
DURATION_OPERATION		INTEGER							
OPERATION_ID		INTEGER							
PATIENT_ID		INTEGER							
ROOM_ID		INTEGER							



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

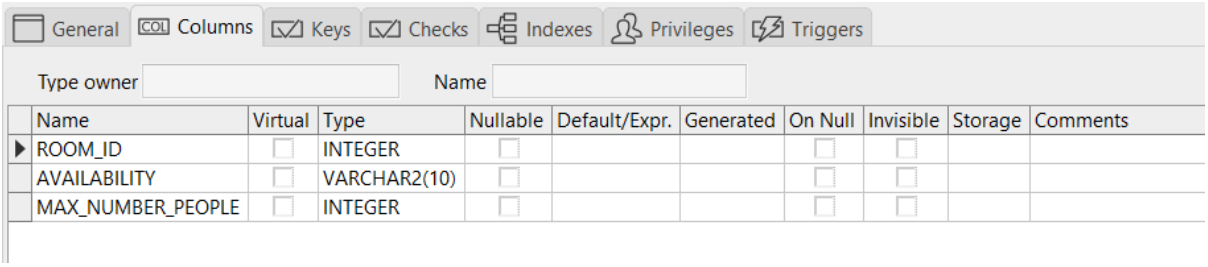
```
-- Create table
create table OPERATION
(
  operation_date     DATE not null,
  duration_operation INTEGER not null,
  operation_id       INTEGER not null,
  patient_id         INTEGER not null,
  room_id            INTEGER not null
)
tablespace SYSTEM
  pctfree 10
  pctused 40
  initrans 1
  maxtrans 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:



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	☑				☐	☐	☑	27/05/2024 20:47:52



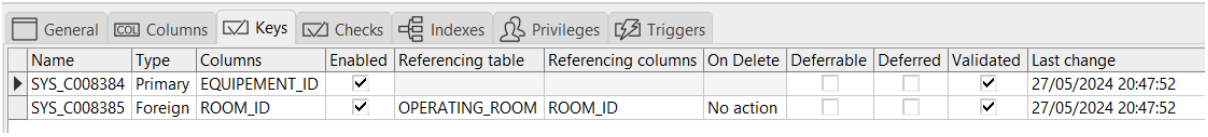
General | Columns | Keys | Checks | Indexes | Privileges | Triggers

Type owner: Name:

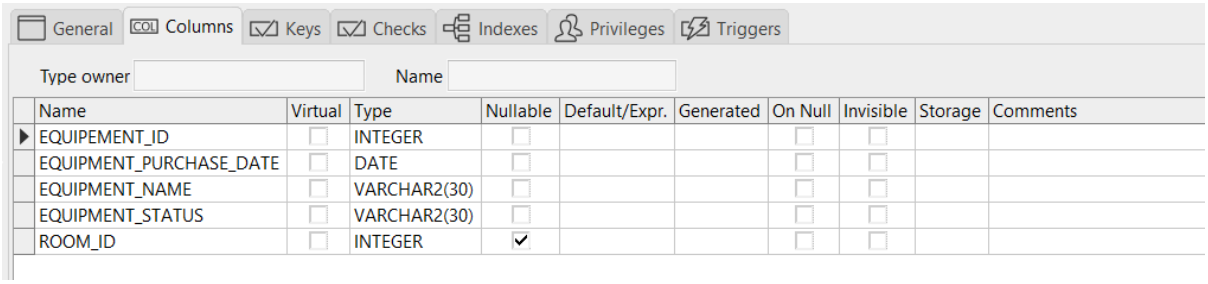
Name	Virtual	Type	Nullable	Default/Expr.	Generated	On Null	Invisible	Storage	Comments
ROOM_ID	☐	INTEGER	☐			☐	☐		
AVAILABILITY	☐	VARCHAR2(10)	☐			☐	☐		
MAX_NUMBER_PEOPLE	☐	INTEGER	☐			☐	☐		

```
-- 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:



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

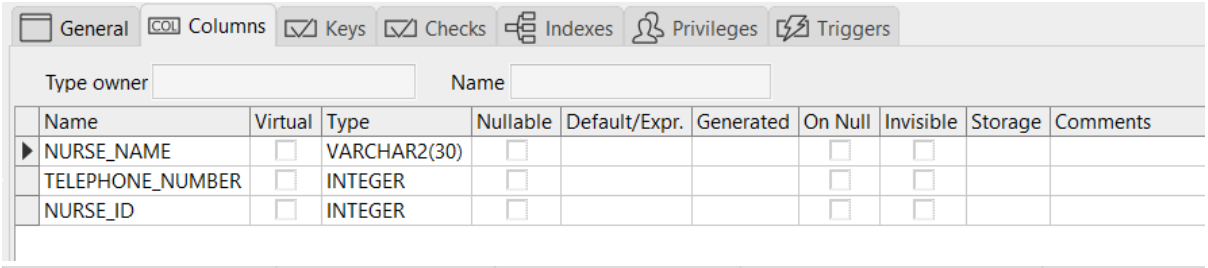


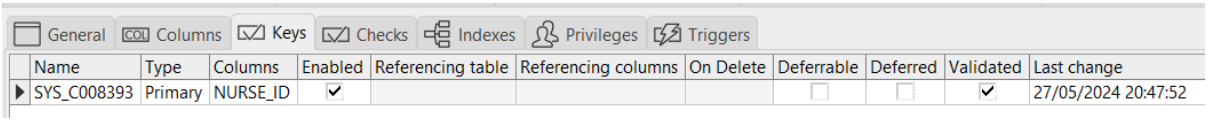
Type owner Name

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
```

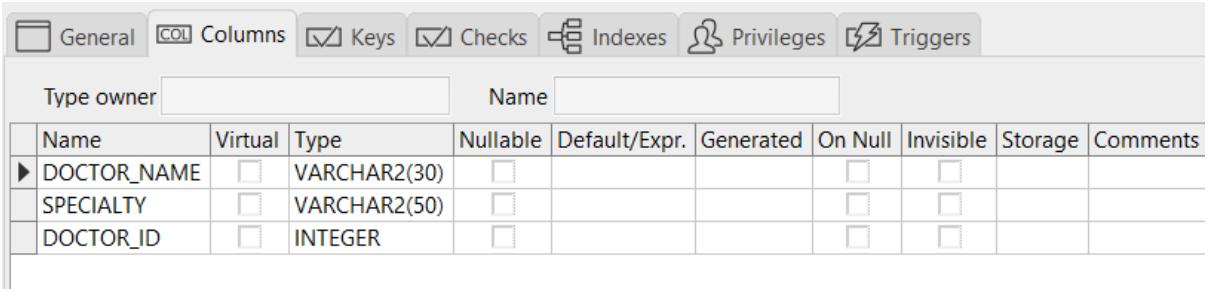
Creating the **Nurse** table:

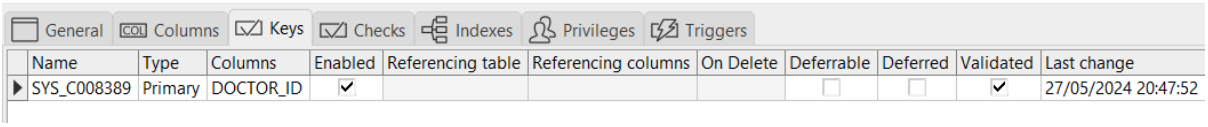





```
-- Create table
create table NURSE
(
  nurse_name       VARCHAR2(30) not null,
  telephone_number INTEGER not null,
  nurse_id         INTEGER not null
)
tablespace SYSTEM
  pctfree 10
  pctused 40
  initrans 1
  maxtrans 255
  storage
  (
    initial 64K
    next 1M
    minextents 1
    maxextents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

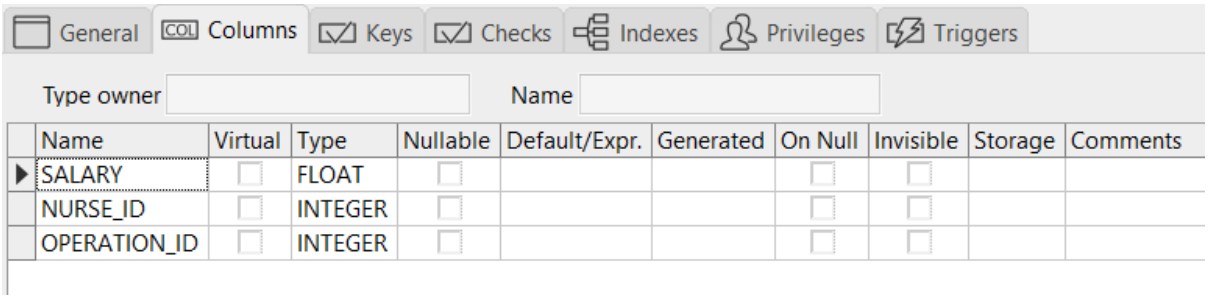
Creating the **Doctor** table:






```
-- 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
```

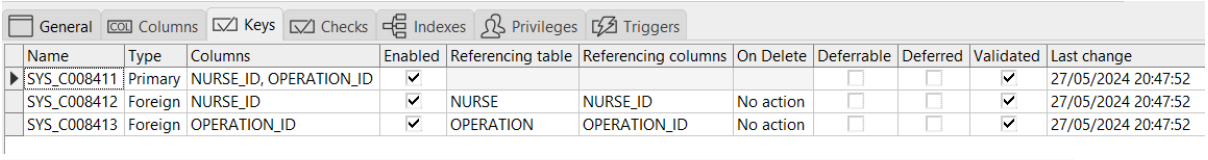
Creating the **Assist by** table:



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							

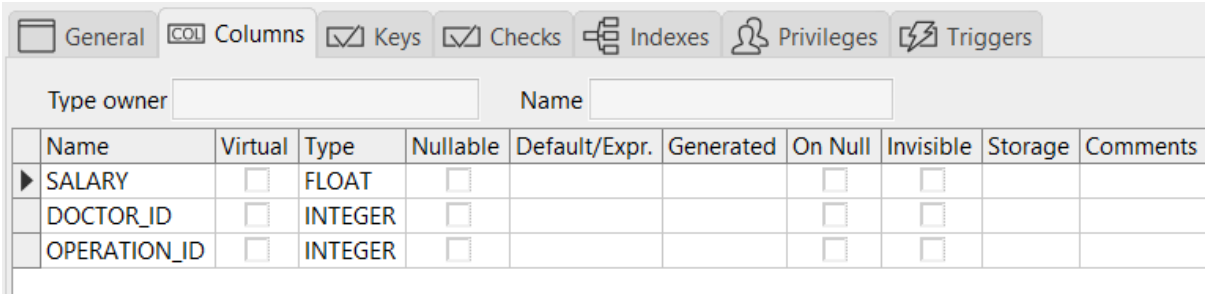


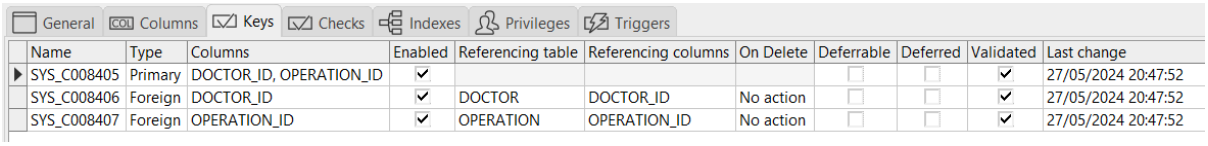
General | Columns | Keys | Checks | Indexes | Privileges | Triggers

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:

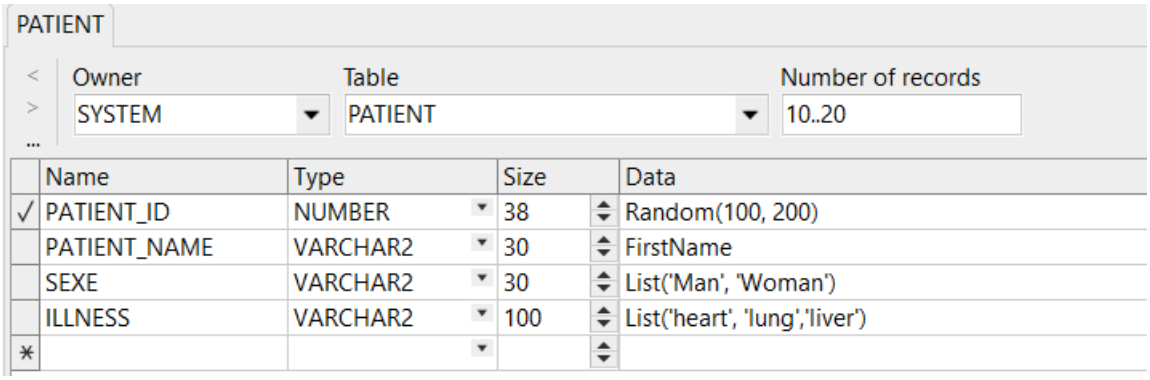





```
-- Create table
create table OPERATE_BY
(
  salary       FLOAT not null,
  doctor_id    INTEGER not null,
  operation_id INTEGER not null
)
tablespace SYSTEM
  pctfree 10
  pctused 40
  initrans 1
  maxtrans 255
  storage
  (
    initial 64K
    next 1M
    minextents 1
    maxextents unlimited
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

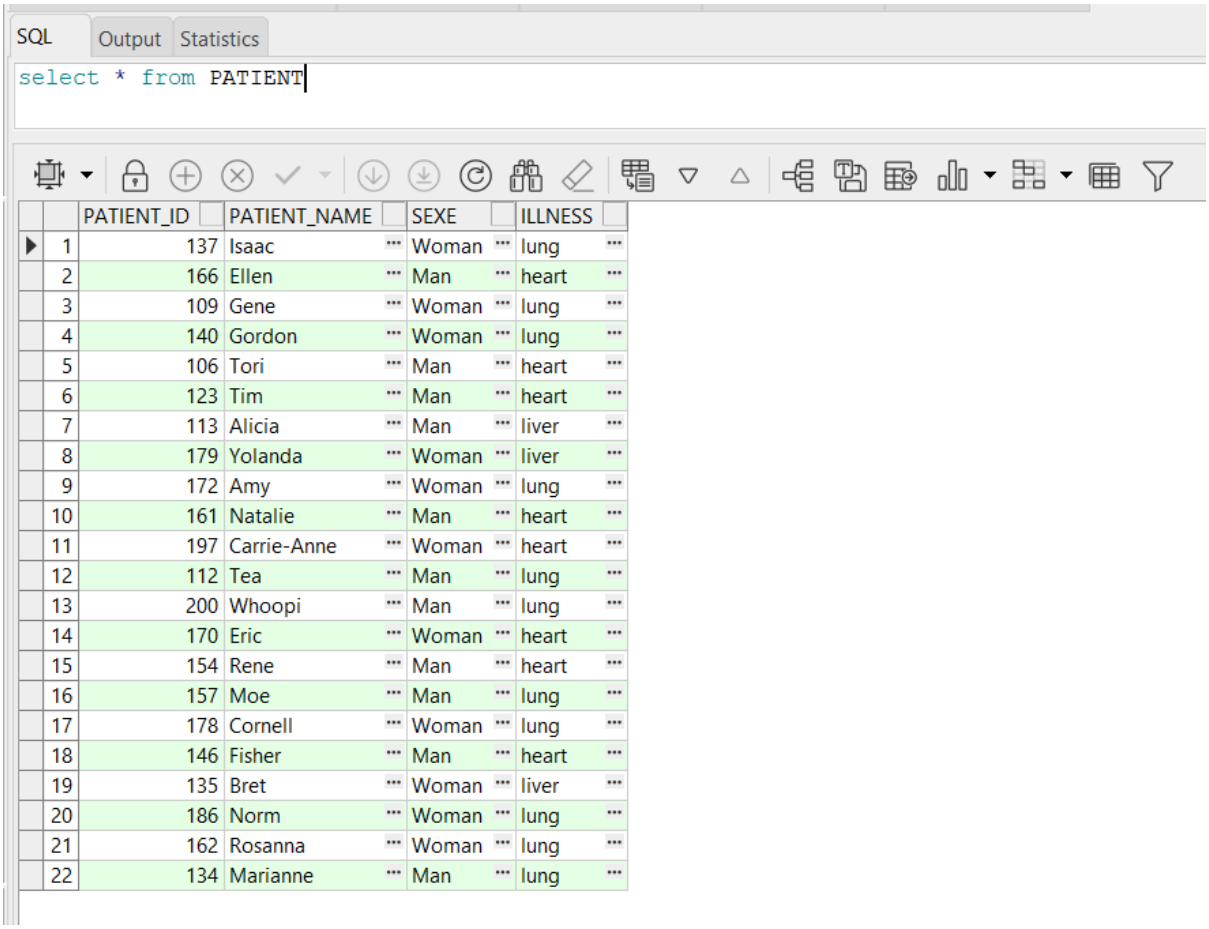
Entering data into the Patient table:



PATIENT

Owner: SYSTEM | Table: PATIENT | Number of records: 10..20

	Name	Type	Size	Data
✓	PATIENT_ID	NUMBER	38	Random(100, 200)
	PATIENT_NAME	VARCHAR2	30	FirstName
	SEXE	VARCHAR2	30	List('Man', 'Woman')
	ILLNESS	VARCHAR2	100	List('heart', 'lung','liver')
*				



SQL | Output | Statistics

```
select * from PATIENT
```

	PATIENT_ID	PATIENT_NAME	SEXE	ILLNESS
1	137	Isaac	Woman	lung
2	166	Ellen	Man	heart
3	109	Gene	Woman	lung
4	140	Gordon	Woman	lung
5	106	Tori	Man	heart
6	123	Tim	Man	heart
7	113	Alicia	Man	liver
8	179	Yolanda	Woman	liver
9	172	Amy	Woman	lung
10	161	Natalie	Man	heart
11	197	Carrie-Anne	Woman	heart
12	112	Tea	Man	lung
13	200	Whoopi	Man	lung
14	170	Eric	Woman	heart
15	154	Rene	Man	heart
16	157	Moe	Man	lung
17	178	Cornell	Woman	lung
18	146	Fisher	Man	heart
19	135	Bret	Woman	liver
20	186	Norm	Woman	lung
21	162	Rosanna	Woman	lung
22	134	Marianne	Man	lung

Entering data into the Operating_Room table:

OPERATING_ROOM

Owner: SYSTEM | Table: OPERATING_ROOM | Number of records: 10..20

Name	Type	Size	Data
ROOM_ID	NUMBER		Random(0, 100)
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	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

Entering data into the Doctor table:

DOCTOR

Owner: SYSTEM | Table: DOCTOR | Number of records: 10..20

Name	Type	Size	Data
DOCTOR_NAME	VARCHAR2	30	FirstName
SPECIALTY	VARCHAR2	50	List('Cardiology', 'Neurology', 'Orthopedic surgery')
DOCTOR_ID	NUMBER		Random(200, 300)

SQL Output Statistics

```
select * from DOCTOR
```

	DOCTOR_NAME	SPECIALTY	DOCTOR_ID
1	Amanda	Cardiology	283
2	Arturo	Orthopedic surgery	268
3	Albert	Orthopedic surgery	296
4	Pelvic	Orthopedic surgery	262
5	Gaby	Cardiology	272
6	Denzel	Cardiology	289
7	Nickel	Neurology	232
8	Kelly	Orthopedic surgery	215
9	Mark	Orthopedic surgery	212
10	Mitchell	Cardiology	297
11	Olympia	Neurology	207
12	Ernest	Neurology	222

Entering data into the Nurse table:

NURSE

Owner: SYSTEM Table: NURSE Number of records: 10..20

Name	Type	Size	Data
NURSE_NAME	VARCHAR2	30	FirstName
TELEPHONE_NUMBER	NUMBER		Random(0000000000, 9999999999)
NURSE_ID	NUMBER		Random(300, 400)

SQL Output Statistics

```
select * from NURSE
```

	NURSE_NAME	TELEPHONE_NUMBER	NURSE_ID
1	Davy	0	379
2	Goldie	0	346
3	Maury	0	328
4	Night	0	314
5	Chrissie	0	394
6	Sheryl	0	352
7	Rebeka	0	319
8	Lindsey	0	390
9	Roy	0	375
10	Suzanne	0	301
11	Lou	0	371
12	Rupert	0	327
13	Catherine	0	349
14	Rutger	0	393
15	Hookah	0	317
16	Natacha	0	357
17	Gwyneth	0	389
18	Talvin	0	351
19	Carla	0	300
20	Barry	0	400
21	Petula	0	320
22	Andre	0	341
23	Pierce	0	348

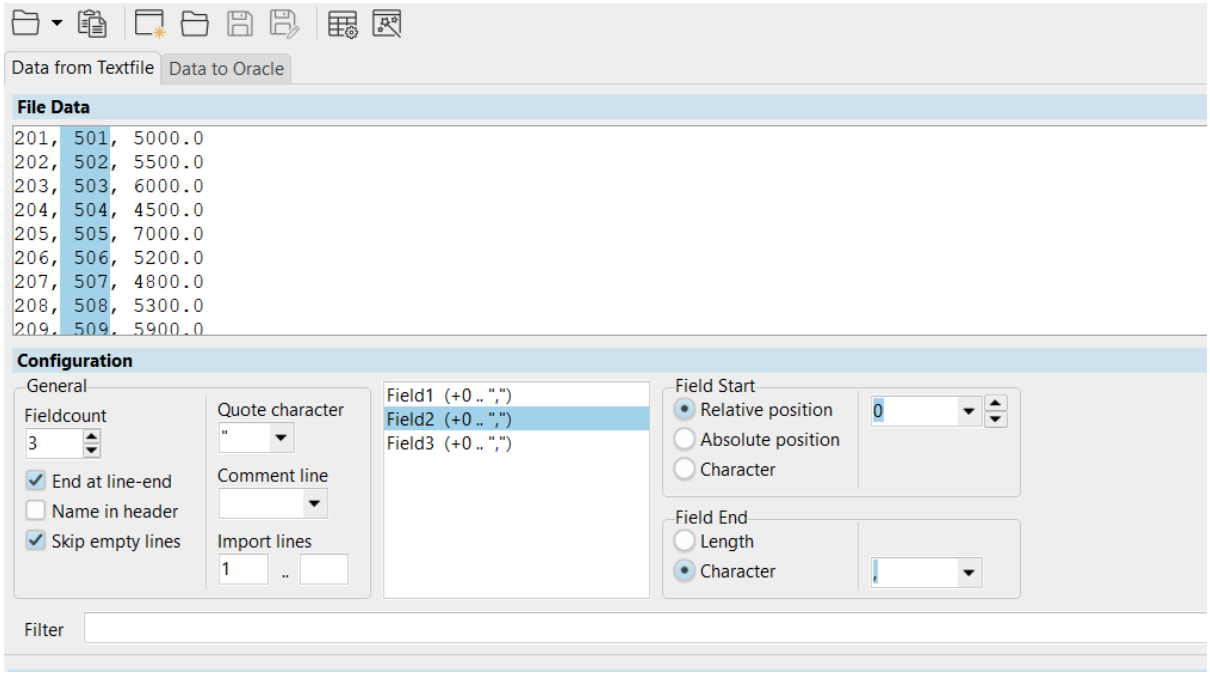
Entering data by TEXT file

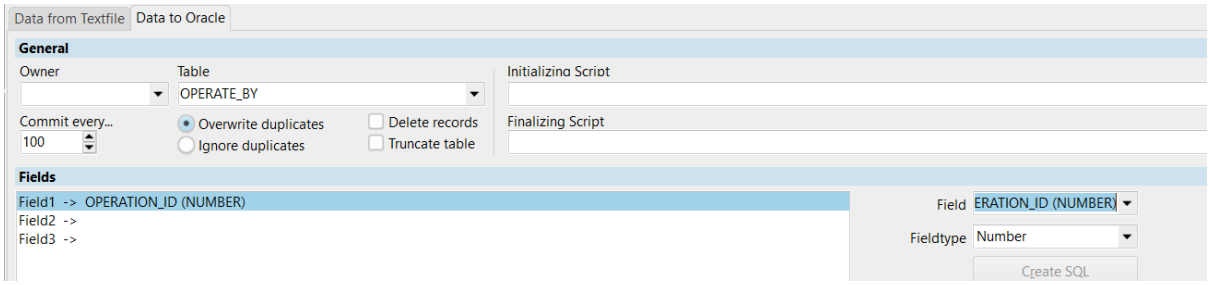
Inserting data into the Operate_by table:

*textimport - Bloc-notes

Fichier Edition Format Affichage Aide

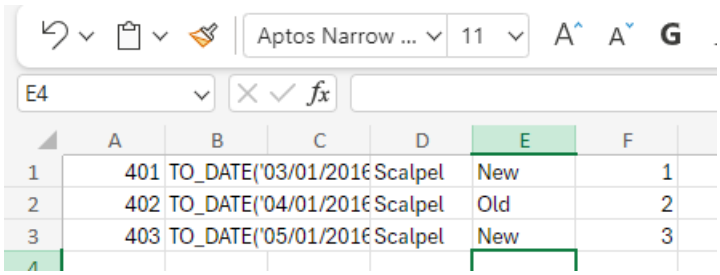
```
201, 501, 5000.0
202, 502, 5500.0
203, 503, 6000.0
204, 504, 4500.0
205, 505, 7000.0
206, 506, 5200.0
207, 507, 4800.0
208, 508, 5300.0
209, 509, 5900.0
210, 510, 6200.0
```

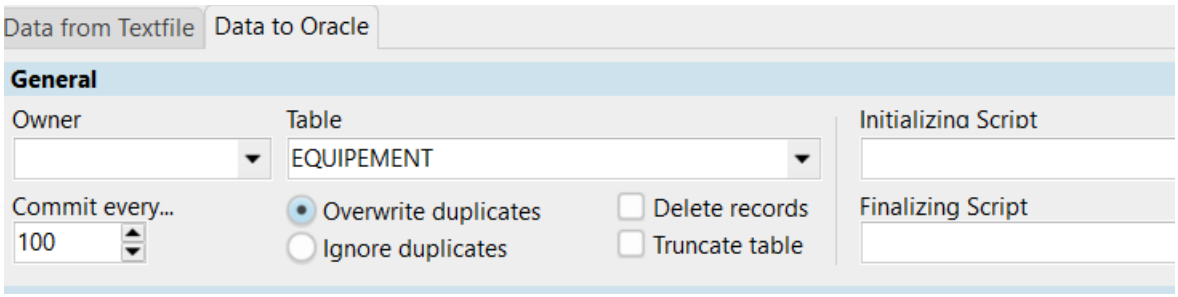


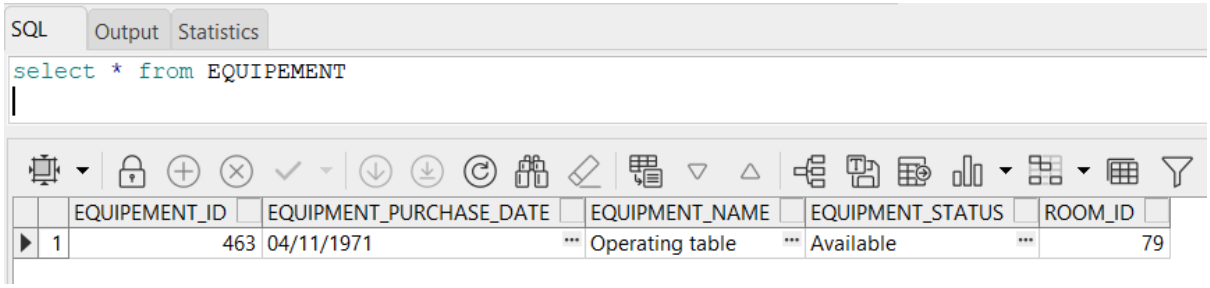


Entering data by EXCEL:

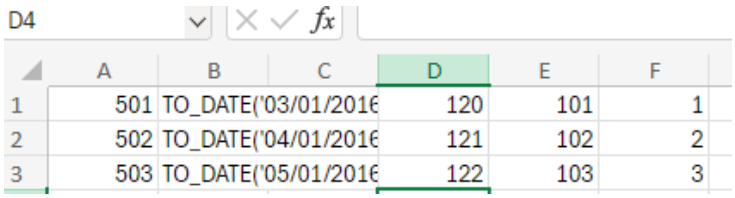
Entering data into the Equipement table:

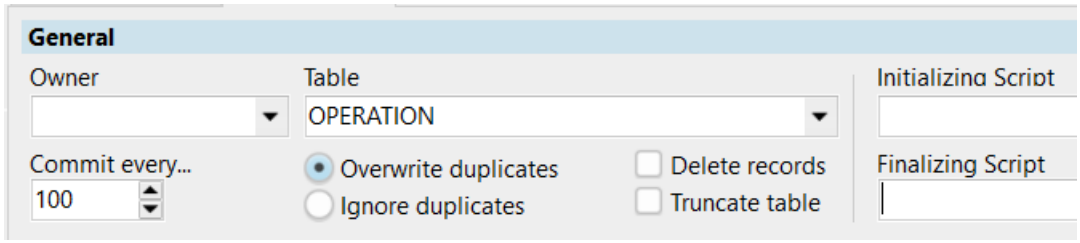






Entering data into the Operation table:





SQL | Output | Statistics

```
select * from OPERATION
```

	OPERATION_DATE	DURATION_OPERATION	OPERATION_ID	PATIENT_ID	ROOM_ID
1	26/12/1955	8	145	135	30
2	06/04/1981	13	183	140	15
3	31/10/2000	15	186	161	67
4	27/05/1943	11	160	170	79
5	19/03/1997	21	169	134	69
6	06/01/2007	3	191	161	69
7	18/09/2004	9	118	135	69
8	09/03/2002	11	192	135	47
9	17/01/1985	14	187	137	43
10	01/03/1941	24	115	154	1
11	29/05/1947	19	165	162	52
12	30/07/1975	19	101	166	47
13	29/08/1960	21	111	162	52
14	27/03/1971	22	116	146	30
15	11/05/1973	7	107	172	69
16	07/07/1953	9	127	109	43
17	17/03/1917	6	158	146	69
18	08/06/1909	4	105	112	1

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
12	2	Available	5