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 (צִיוד):

- 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)
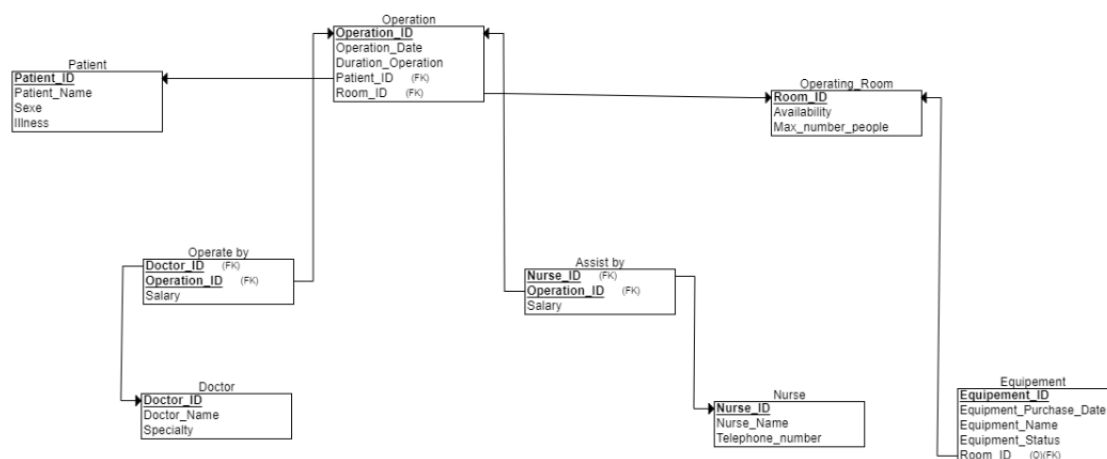An operation concerns a single patient.

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:
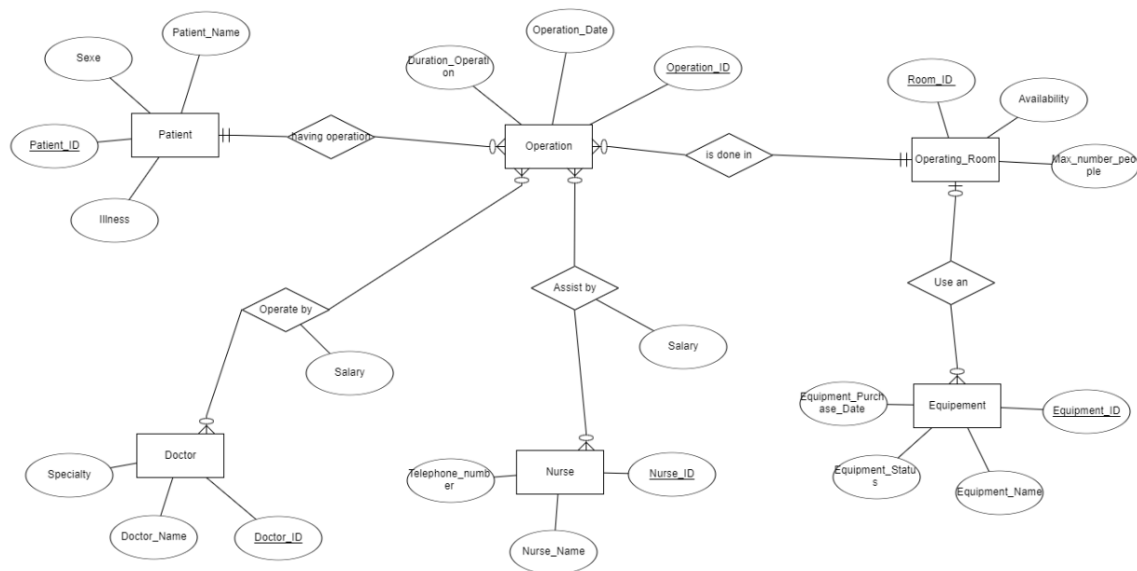
DSD diagram:

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	COL 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	COL 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	☐			☐	☐		

General | COL 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:

```sql
-- 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

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:

Type owner [   ]   Name [   ]

Name	Virtual	Type	Nullable	Default/Expr.	Generated	On Null	Invisible	Storage	Comments
▶ NURSE_NAME	☐	VARCHAR2(30)	☐			☐	☐		
TELEPHONE_NUMBER	☐	INTEGER	☐			☐	☐		
NURSE_ID	☐	INTEGER	☐			☐	☐		

Name	Type	Columns	Enabled	Referencing table	Referencing columns	On Delete	Deferrable	Deferred	Validated	Last change
▶ SYS_C008393	Primary	NURSE_ID	✔				☐	☐	✔	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:

Name	Virtual	Type	Nullable	Default/Expr.	Generated	On Null	Invisible	Storage	Comments
▶ DOCTOR_NAME	☐	VARCHAR2(30)	☐			☐	☐		
SPECIALTY	☐	VARCHAR2(50)	☐			☐	☐		
DOCTOR_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_C008389	Primary	DOCTOR_ID	☑				☐	☐	☑	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:

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

| | General | COL 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:

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

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
```

	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_PURCHAS	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',
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:

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
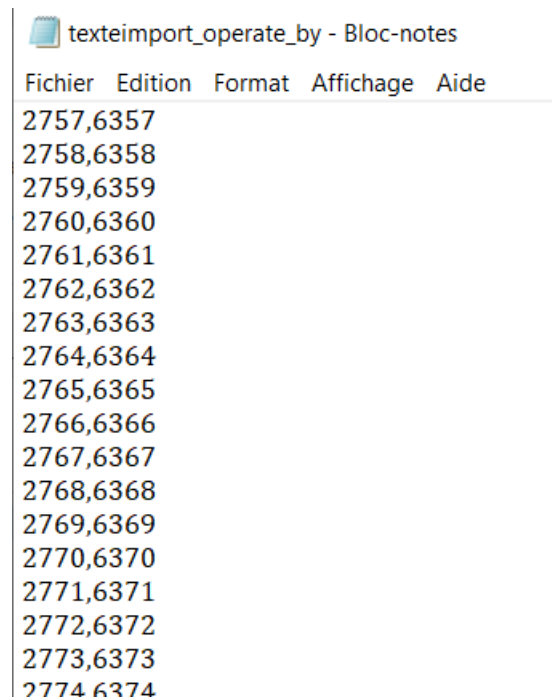
```
select * from Nurse
```

	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:

texteimport_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   `0`
○ Absolute position
○ Character

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

```
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:

📄 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
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

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

Quote character
"

Comment line

Import lines
1 ..

Field1 (+0 .. ",") Nurse_ID
Field2 (+0 .. ",") Operation

Filter

Result Preview

Nurse_ID	Operation_ID
3600	6000
3601	6001

Data from Textfile | **Data to Oracle**

General

Owner
C##ELIAOU_HAIM ▾

Table
ASSIST_BY

Commit every...
100 ▴▾

◉ Overwrite duplicates
○ Ignore duplicates

☐ D
☐ Tr

Fields

Field1 Nurse_ID -> NURSE_ID
Field2 Operation_ID -> OPERATION_ID

Result Preview

Nurse_ID	Operation_ID
3600	6000
3601	6001

SQL   Output   Statistics

```sql
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
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
08/06/2024,2,6014,114,1214

MOCK DATA OPERATION

Data from Textfile | Data to Oracle

General

Owner: C##ELIAOU_HAIM
Table: OPERATION
Initializing Script

Commit every... 100
( ) Overwrite duplicates
( ) Ignore duplicates
[ ] Delete records
[ ] Truncate table
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 exampl

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@XE [17:49:50] 465 records imported in 25.094 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
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

```sql
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: