Normalization And Schema Refinement

For

Vaccination or Other Health Drives Database

Prepared By

Hiten Padaliya (201901401)
Satyam Bhut (201901408)
Dhairya Rupala (201901419)
Jenish Rathod (201901435)

Organization

Dhirubhai Ambani Institute of Information and Communication Technology

Original Design of Database

Relational Schema

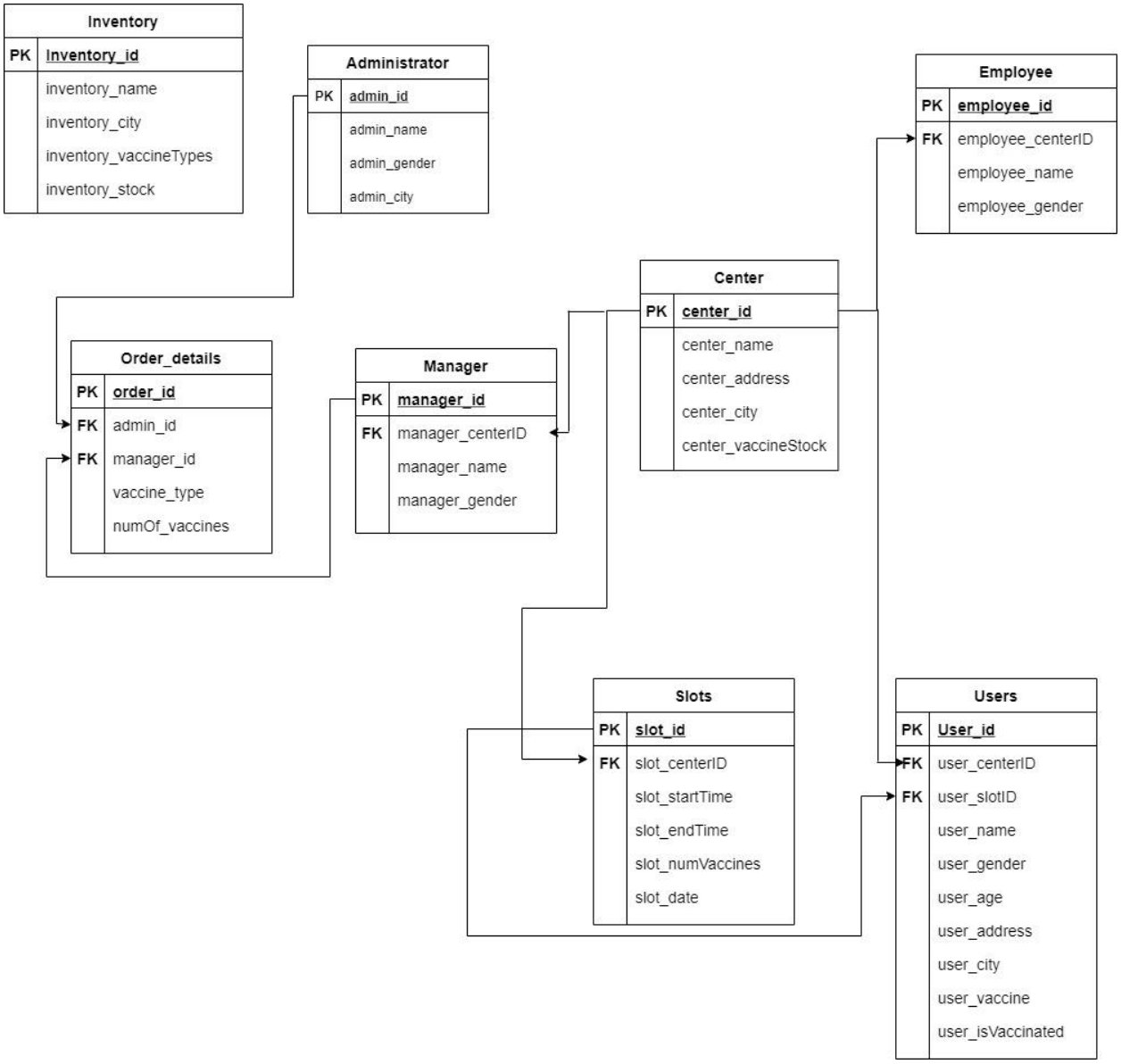


Table Schema

Inventory(Inventory_id, inventory_name, inventory_city, inventory_vaccineTypes, inventory_stock**)**

Administrator(admin_id, admin_name ,admin_gender, admin_city)

Center(center_id, center_name, center_address, center_city, center_vaccineStock)

Slots(slot_id, slot_centerID, slot_startTime, slotendTime, slot_numVaccines, slot_date)

Manager(manager_id, manager_centerID, manager_name, manager_gender)

Order_details(order_id, admin_id, manager_id, vaccineType, numOf_vaccines)

Employee(employee_id, employee_centerID, employee_name, employee_gender)

Users(user_id, user_centerID, user_slotID, user_name, user_gender, user_age, user_address, user_city, user_vaccine, user_isVaccinated)

Documentation of Normalization & Schema Refinement

Functional Dependencies:

Administrator
Primary Key = admin_id
Admin_id -> admin_name
Admin_id -> admin_gender
Admin_id -> admin_city

Center
Primary key = center_id
Center_id -> center_name
Center_id -> center_address
Center_id -> center_city
Center_id -> center_vaccineStock

Employee
Primary key = employee_id
Foreign key = employee_centerID
Employee_id -> employee_name
Employee_id -> employee_gender

Manager
Primary key = manager_id
Foreign key = manager_centerID
Manager_id -> manager_name
Manager_id -> manager_gender

Order details
Primary key = order_id
Foreign key = admin_id, manager_id
Order_id -> vaccine_type
Order_id -> numOf_vaccines

Inventory

Primary key = Inventory_id

Inventory_id -> inventory_name
Inventory_id -> inventory_city
Inventory_id -> inventory_vaccineTypes
Inventory_id -> inventory_stock

Slots

Primary key = slot_id

Foreign key = slot_centerID

Slot_id -> slot_startTime
Slot_id -> slot_endTime
Slot_id -> slot_numVaccines
Slot_id -> slot_date

Users

Primary key = user_id

Foreign key = user_centerID, user_slotID

User_id -> user_name
User_id -> user_gender
User_id -> user_age
User_id -> user_address
User_id -> user_city
User_id -> user_vaccine
User_id -> user_isVaccinated

Normalization to 1NF:
In 1NF form in each and every relation of the Database, there should not be any repeating columns within a row or not multivalued columns.

Here, in our database design, all the listed attributes in the relations are atomic, also wherever we have the address attribute in the relation, we stored the address in a single string. We are also considering only one entry of each of the attributes. So the above database design is the 1NF normalized form.

Normalization to 2NF:
In the 2NF normalized form, each and every relationship should be in the 1NF form also there should not be any partial dependencies in the relations. Partial dependency means that no nonprime attributes are dependent on any proper subset of the candidate key of the relation.

For any database design relations with only one Primary key (not composite) are already in the 2NF form.

Here from the above functional dependencies, it is clear that this database design has no composite primary keys, hence no partial dependencies. So this design is already in the 2NF normalized form.

Normalization to 3NF
For the relations to be in the 3NF, all the relations at first have to be 2NF form, and for the 3NF there should not be any transitive functional dependencies in the relations for the non-prime attributes. So there are not any functional dependencies such as non-prime -> non-prime. So this database design is already in the 3NF normalized form.

Normalization to BCNF
For the relation to be in BCNF form, iff all the functional dependencies have a super key as its left-hand side, no non-prime attribute can be on the left-hand side. In our database, all the functional dependencies have a primary key(i.e prime attribute) as their left-hand side. Also, all the foreign keys included in all the relations are the primary keys of the referenced tables.

Update, Delete Anomalies

Relation/Schema	**Update**	**Delete**
Inventory	It'll not affect the relation	It'll not affect the relation
Administrator	It'll update the foreign keyed entries of the order_detail relation.	It'll delete the foreign keyed entries of the order_detail relation.
Center	It'll update the slot, employee, manager, and user relations.	It'll delete the foreign keyed entries of all the relations except users.
Slots	It'll update the users foreign keyed entries.	It'll set the null value to the foreign keyed entries of the users relation.
Manager	It'll update the foreign keyed entries of the order_detail relation.	It'll delete the foreign keyed entries of the order_detail relation.
Order_details	It'll not affect the relation	It'll not affect the relation
Employee	It'll not affect the relation	It'll not affect the relation
Users	It'll not affect the relation	It'll not affect the relation

NOTE - Here there are no insertion anomalies in any of the relations.

DDL Scripts:

- **Inventory Table**

```
CREATE TABLE Inventory
(
        inventory_id int not null,
        inventory_name varchar(100) not null,
        inventory_city varchar(100) not null,
        inventory_vaccineTypes varchar(200) not null,
        inventory_stock Int not null default 0,
        primary key (inventory_id),
        check(inventory_stock>=0)
);
```

- **Administrator Table**

```
CREATE TABLE Administrator
(
        admin_id int not null,
        admin_name varchar(50) not null,
        admin_gender varchar(6) not null,
        admin_city varchar(30) not null,
        primary key (admin_id),
        check(admin_gender in ('Female','Male'))
);
```

- **Center Table**

```
CREATE TABLE Center
(
        center_id int not null,
        center_name varchar(50) not null,
        center_address varchar(200) not null,
        center_city varchar(30) not null,
        center_vaccineStock int not null default 0,
        primary key (center_id),
        check(center_vaccineStock>=0)
);
```

- **Slots Table**

```
CREATE TABLE Slots
(
        slot_id int not null,
        slot_centerID int,
        slot_startTime varchar not null default '00:00',
        slot_endTime varchar not null default '00:00',
        slot_numVaccines int not null default 0,
        Slot_date varchar not null default '00.00.0000',
        check (slot_numVaccines>=0),
        primary key (slot_id),
        foreign key (slot_centerID) references Center(center_id) on delete cascade on update cascade
);
```

- **Manager Table**

```
CREATE TABLE Manager
(
        manager_id int not null,
        manager_centerID int,
        manager_name varchar(50) not null,
        manager_gender varchar(6) not null,
        primary key (manager_id),
        foreign key (manager_centerID) references Center(center_id) on delete cascade on update cascade,
        check(manager_gender in ('Female','Male'))
);
```

- **Order_details Table**

```
CREATE TABLE Order_details
(
        order_id int not null,
        admin_id int,
        manager_id int,
        vaccineType varchar(200),
        numOf_vaccines int default 0,
        primary key (order_id),
```

foreign key (admin_id) references administrator(admin_id) on delete cascade on update cascade,

foreign key (manager_id) references manager(manager_id) on delete cascade on update cascade);

- **Employee Table**

CREATE TABLE Employee
(
employee_id **int not** null,
employee_centerID **int**,
employee_name **varchar(50) not** null,
employee_gender **varchar(6) not** null,
primary key (employee_id),
foreign key (employee_centerID) **references** Center(center_id) **on delete cascade on update cascade,**
check(employee_gender **in** ('Female','Male'))
);

- **Users Table**

CREATE TABLE Users
(
user_id **int** not null,
user_centerID **int**,
user_slotID **int**,
user_name **varchar(50) not** null,
user_gender **varchar(6**) **not** null,
user_age **int**,
user_address **varchar(200) not** null,
user_city **varchar(30) not** null,
user_vaccine **varchar(200) default** null,
user_isVaccinated **char(1) default** 'N',

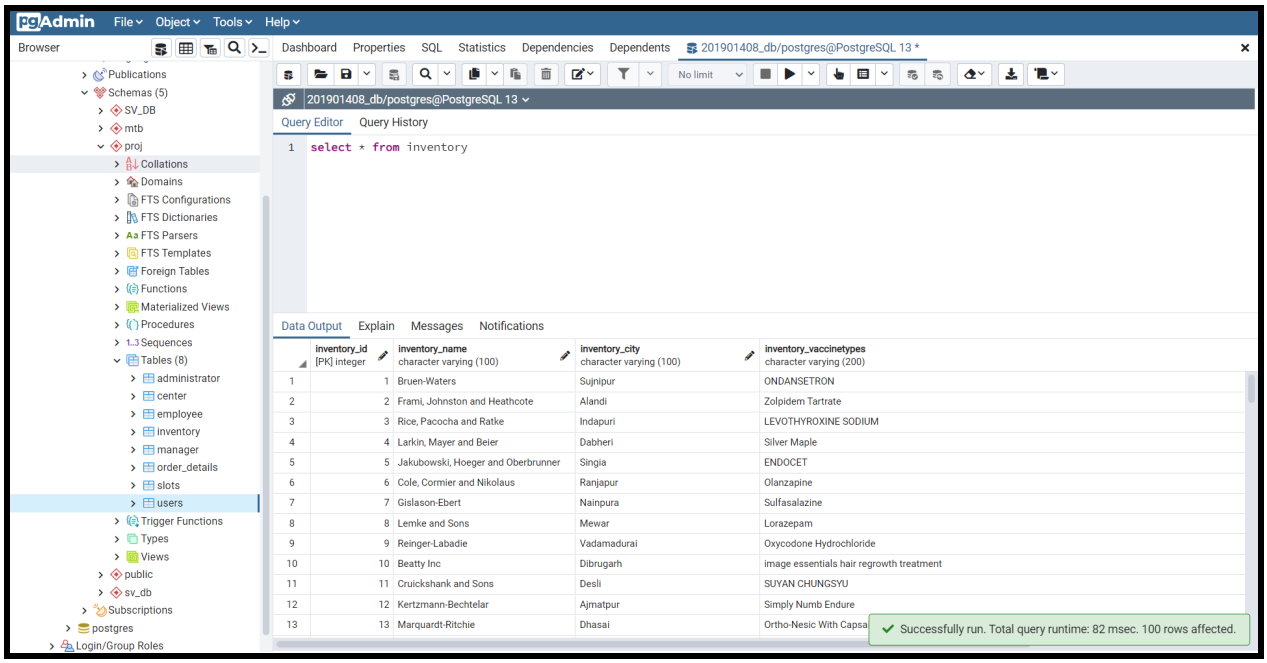
primary key (user_id),
foreign key (user_centerID) **references** Center(center_id) **on delete set null on update cascade,**

```
        foreign key (user_slotID) references Slots(slot_id) on delete set null on update
cascade,
        check(user_gender in ('Female','Male')),
        check(user_age>0),
        check (user_isVaccinated in ('Y','N','y','n'))
    );
```

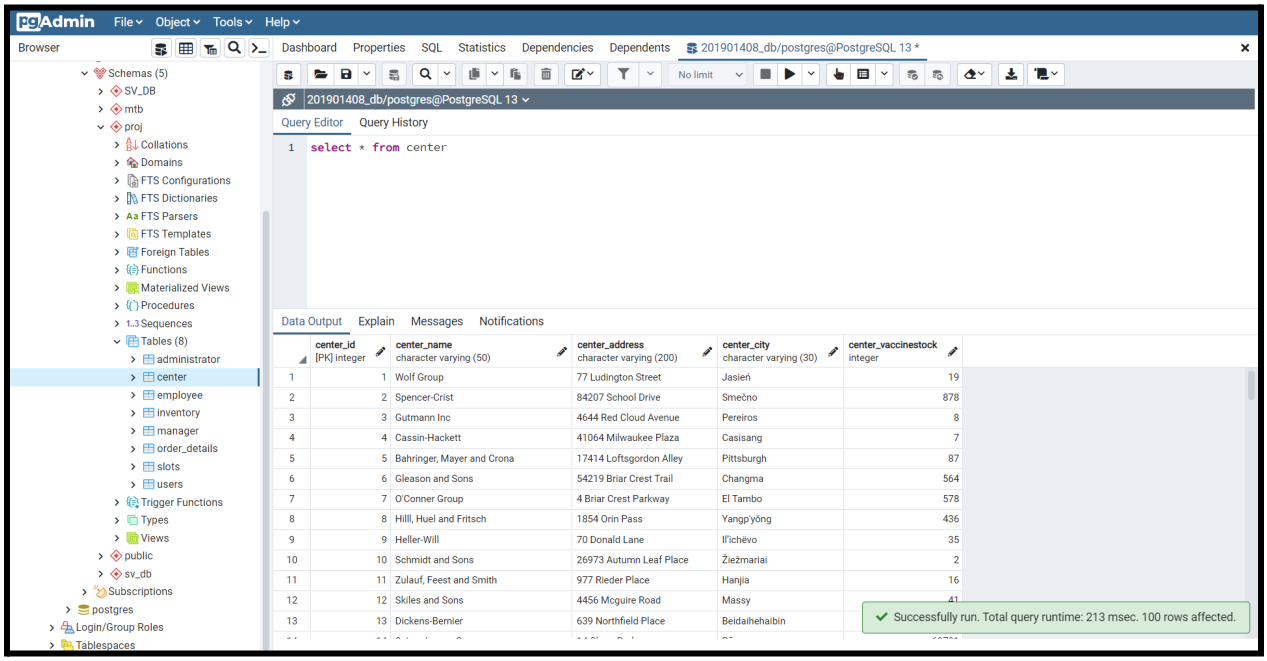
Creation of Tables

.

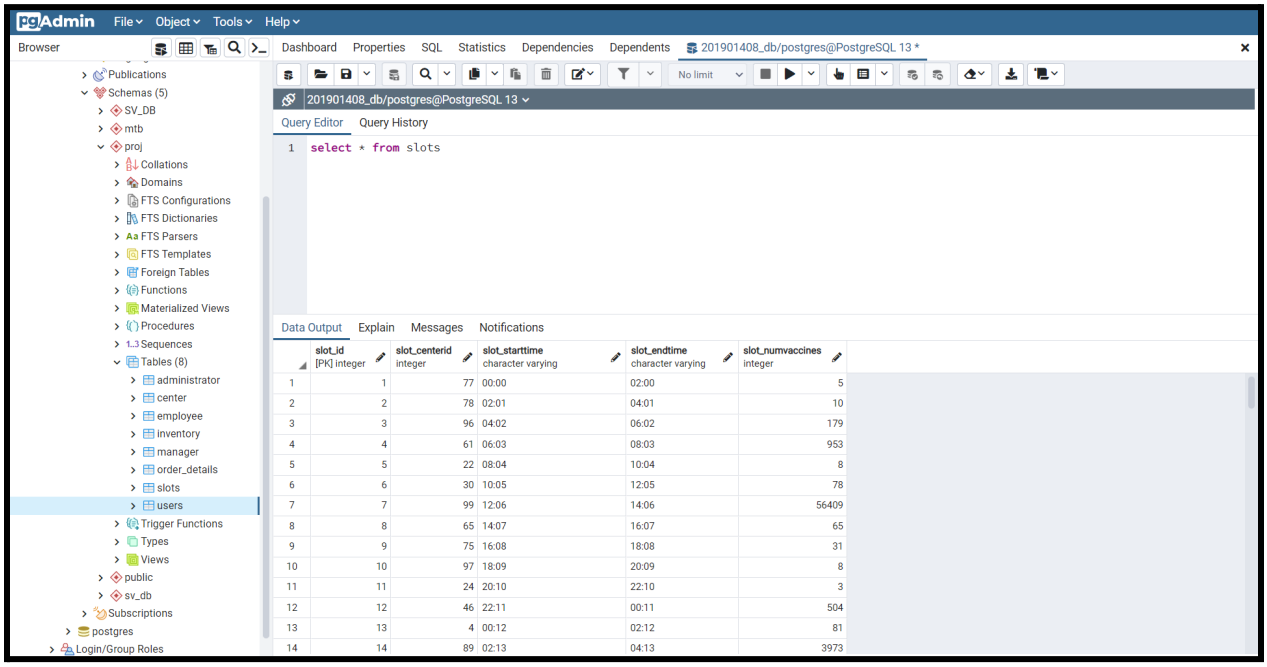
Inventory Data


pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents 201901408_db/postgres@PostgreSQL 13 *
201901408_db/postgres@PostgreSQL 13
Query Editor Query History
1 select * from inventory
Data Output Explain Messages Notifications
inventory_id [PK] integer
inventory_name character varying (100)
inventory_city character varying (100)
inventory_vaccinetypes character varying (200)
1 1 Bruen-Waters Sujnipur ONDANSETRON
2 2 Frami, Johnston and Heathcote Alandi Zolpidem Tartrate
3 3 Rice, Pacocha and Ratke Indapuri LEVOTHYROXINE SODIUM
4 4 Larkin, Mayer and Beier Dabheri Silver Maple
5 5 Jakubowski, Hoeger and Oberbrunner Singia ENDOCET
6 6 Cole, Cormier and Nikolaus Ranjapur Olanzapine
7 7 Gislason-Ebert Nainpura Sulfasalazine
8 8 Lemke and Sons Mewar Lorazepam
9 9 Reinger-Labadie Vadamadurai Oxycodone Hydrochloride
10 10 Beatty Inc Dibrugarh image essentials hair regrowth treatment
11 11 Cruickshank and Sons Desli SUYAN CHUNGSYU
12 12 Kertzmann-Bechtelar Ajmatpur Simply Numb Endure
13 13 Marquardt-Ritchie Dhasai Ortho-Nesic With Capsa
Successfully run. Total query runtime: 82 msec. 100 rows affected.
Publications
Schemas (5)
SV_DB
mtb
proj
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Procedures
Sequences
Tables (8)
administrator
center
employee
inventory
manager
order_details
slots
users
Trigger Functions
Types
Views
public
sv_db
Subscriptions
postgres
Login/Group Roles

Center Data

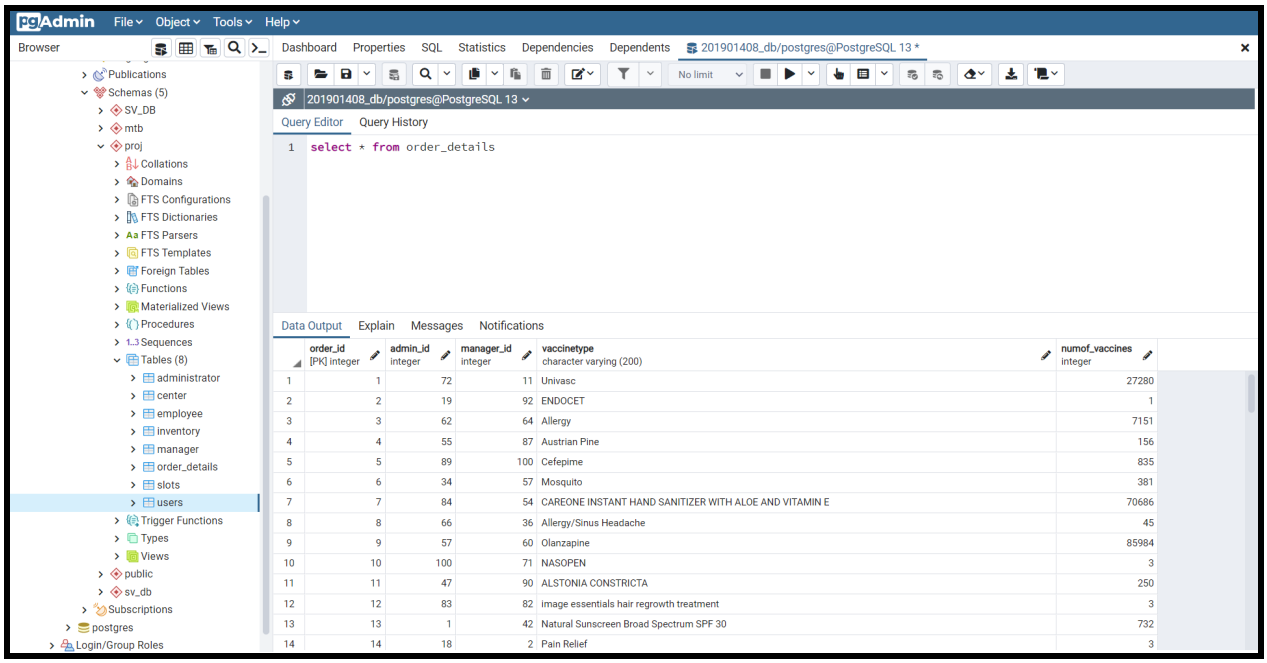

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents 201901408_db/postgres@PostgreSQL 13 *
201901408_db/postgres@PostgreSQL 13
Query Editor Query History
1 select * from center
Data Output Explain Messages Notifications
center_id [PK] integer
center_name character varying (50)
center_address character varying (200)
center_city character varying (30)
center_vaccinestock integer
1 1 Wolf Group 77 Ludington Street Jasień 19
2 2 Spencer-Crist 84207 School Drive Smečno 878
3 3 Gutmann Inc 4644 Red Cloud Avenue Pereiros 8
4 4 Cassin-Hackett 41064 Milwaukee Plaza Casisang 7
5 5 Bahringer, Mayer and Crona 17414 Loftsgordon Alley Pittsburgh 87
6 6 Gleason and Sons 54219 Briar Crest Trail Changma 564
7 7 O'Conner Group 4 Briar Crest Parkway El Tambo 578
8 8 Hilll, Huel and Fritsch 1854 Orin Pass Yangp'yŏng 436
9 9 Heller-Will 70 Donald Lane Il'ichëvo 35
10 10 Schmidt and Sons 26973 Autumn Leaf Place Žiežmariai 2
11 11 Zulauf, Feest and Smith 977 Rieder Place Hanjia 16
12 12 Skiles and Sons 4456 Mcguire Road Massy 41
13 13 Dickens-Bernier 639 Northfield Place Beidaihehaibin
Successfully run. Total query runtime: 213 msec. 100 rows affected.
Schemas (5)
SV_DB
mtb
proj
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Procedures
Sequences
Tables (8)
administrator
center
employee
inventory
manager
order_details
slots
users
Trigger Functions
Types
Views
public
sv_db
Subscriptions
postgres
Login/Group Roles
Tablespaces

Slots Data

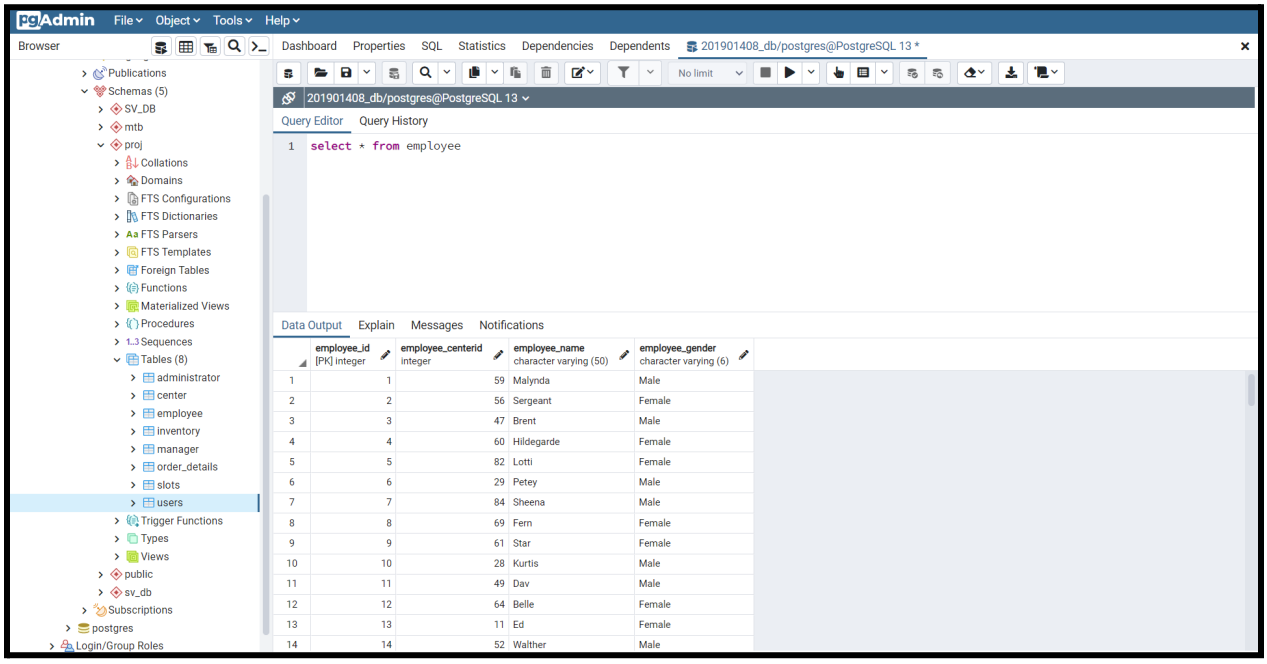


	slot_id [PK] integer	slot_centerid integer	slot_starttime character varying	slot_endtime character varying	slot_numvaccines integer
1	1	77	00:00	02:00	5
2	2	78	02:01	04:01	10
3	3	96	04:02	06:02	179
4	4	61	06:03	08:03	953
5	5	22	08:04	10:04	8
6	6	30	10:05	12:05	78
7	7	99	12:06	14:06	56409
8	8	65	14:07	16:07	65
9	9	75	16:08	18:08	31
10	10	97	18:09	20:09	8
11	11	24	20:10	22:10	3
12	12	46	22:11	00:11	504
13	13	4	00:12	02:12	81
14	14	89	02:13	04:13	3973

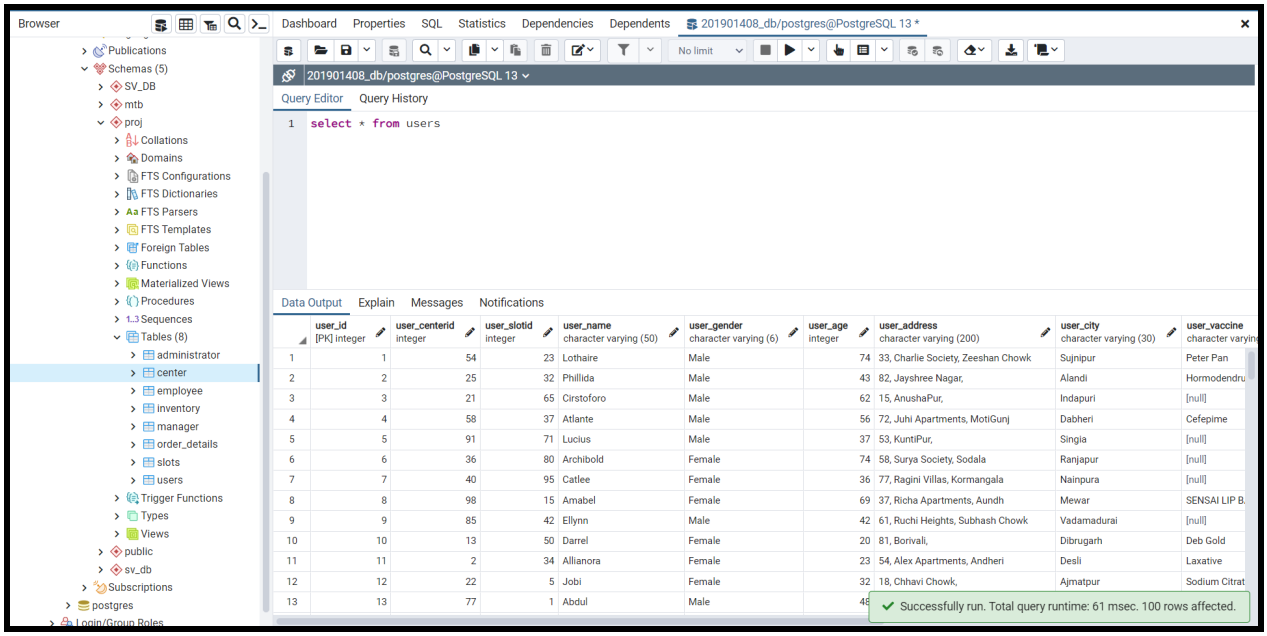
Order Details Data



Employee Data



Users Data



Final Table Schema

Inventory(Inventory_id, inventory_name, inventory_city, inventory_vaccineTypes, inventory_stock**)**

Administrator(admin_id, admin_name ,admin_gender, admin_city)

Center(center_id, center_name, center_address, center_city, center_vaccineStock)

Slots(slot_id, slot_centerID, slot_startTime, slotendTime, slot_numVaccines)

Manager(manager_id, manager_centerID, manager_name, manager_gender)

Order_details(order_id, admin_id, manager_id, vaccineType, numOf_vaccines)

Employee(employee_id, employee_centerID, employee_name, employee_gender)

Users(user_id, user_centerID, user_slotID, user_name, user_gender, user_age, user_address, user_city, user_vaccine, user_isVaccinated)