

LAB 8

Normalization and Schema Refinement

Thakkar Amit (201901038)

Makwana Jigar (201901428)

Prajapati Parth (201901429)

Gohil Arpit (201901471)

- ❖ person(**id**, first_name, middle_name, last_name, contact_no, age, email_address, user_type, username, password)
 - Primary key : id
 - Partial dependency : None
 - Transitive dependency: None

 - Insert Anomaly: Nothing found
 - (Attributes and Reason)
 - Update Anomaly: Nothing found
 - Delete Anomaly: Nothing found

 - 1NF : There is no multi-valued attribute in this relation. So this relation is already in 1NF form.
 - 2NF : There is no partial dependency in this relation. So this relation is already in 2NF form.
 - 3NF : There is no transitive dependency in this relation. So this relation is already in 3NF form.
 - BCNF : All functional dependencies are on Primary key (id). So, This relation is already in BCNF form.

❖ roles(**r_name**, work_hours, salary)

- Primary key : r_name
- Partial dependency: None
- Transitive dependency: None

- Insert Anomaly: Nothing found
- Update Anomaly: Nothing found
- Delete Anomaly: Nothing found

- 1NF: There is no multi-valued attribute in this relation. So this relation is already in 1NF form.
- 2NF: There is no partial dependency in this relation. So this relation is already in 2NF form.
- 3NF: There is no transitive dependency in this relation. So this relation is already in 3NF form.
- BCNF: This relation is already in BCNF form.

❖ player(**player_id**, height ,weight ,team_id)

- Primary key : player_id
- Foreign key : player_id(player), team_id(team)
- Partial dependency : None
- Transitive dependency: None

- Insert Anomaly: Nothing found
- (Attributes and Reason)
- Update Anomaly: Nothing found
- Delete Anomaly: Nothing found

- 1NF : There is no multi-valued attribute in this relation. So this relation is already in 1NF form.
- 2NF : There is no partial dependency in this relation. So this relation is already in 2NF form.
- 3NF : There is no transitive dependency in this relation. So this relation is already in 3NF form.
- BCNF : This relation is already in BCNF form.

❖ account(**account_no**, player_id, debit_card_no, date_of_joining, balance)

- Primary key : account_no
- Foreign key : player_id(player)
- Partial dependency: None
- Transitive dependency: None

- Insert Anomaly: Nothing found
- Update Anomaly: Nothing found
- Delete Anomaly: Nothing found

- 1NF: There is no multi-valued attribute in this relation. So this relation is already in 1NF form.
- 2NF: There is no partial dependency in this relation. So this relation is already in 2NF form.
- 3NF: There is no transitive dependency in this relation. So this relation is already in 3NF form.
- BCNF: This relation is already in BCNF form.

❖ game(**game_id**, game_name, type, mode, price, reward, age, height, weight)

- Primary key : game_id
- Foreign key : None
- Partial Dependencies: None

- Transitive Dependencies: None
- Insert Anomaly: Nothing found
- Update Anomaly: Nothing found
- Delete Anomaly: Nothing found
- 1NF : There is no multi-valued attribute in this relation. So this relation is already in 1NF form.
- 2NF : There is no partial dependency in this relation. So this relation is already in 2NF form.
- 3NF : There is no transitive dependency in this relation. So this relation is already in 3NF form.
- BCNF : This relation is already in BCNF form.

❖ game_table(**table_no**, game_id, capacity)

- Primary key : table_no
- Foreign key : game_id (game)
- Partial dependency: None
- Transitive dependency: None
- Insert Anomaly: Nothing found
- Update Anomaly: Nothing found
- Delete Anomaly: Nothing found
- 1NF: There is no multi-valued attribute in this relation. So this relation is already in 1NF form.
- 2NF: There is no partial dependency in this relation. So this relation is already in 2NF form.
- 3NF: There is no transitive dependency in this relation. So this relation is already in 3NF form.
- BCNF: This relation is already in BCNF form.

❖ GameZone_balance(account_no, balance)

- Partial dependency: None
- Transitive dependency: None

- Insert Anomaly: Nothing found
- Update Anomaly: Nothing found
- Delete Anomaly: Nothing found

- 1NF: There is no multi-valued attribute in this relation. So this relation is already in 1NF form.
- 2NF: There is no partial dependency in this relation. So this relation is already in 2NF form.
- 3NF: There is no transitive dependency in this relation. So this relation is already in 3NF form.
- BCNF: This relation is already in BCNF form.

❖ purchase(**purchase_id**, player_id, game_id, no_hour)

- Primary key : purchase_id
- Foreign key : game_id(game), player_id(player)
- Partial Dependencies: None
- Transitive Dependencies: None

- Insert Anomaly: Nothing found
- Update Anomaly: Nothing found
- Delete Anomaly: Nothing found

- 1NF : There is no multi-valued attribute in this relation. So this relation is already in 1NF form.

- 2NF : There is no partial dependency in this relation. So this relation is already in 2NF form.
- 3NF : There is no transitive dependency in this relation. So this relation is already in 3NF form.
- BCNF : This relation is already in BCNF form.

❖ transactions(**transaction_id**, account_no, date_and_time, amount, type, method)

- Primary key : transaction_id
- Foreign key : account_no(account)
- Partial Dependencies: None
- Transitive Dependencies: None

- Insert Anomaly: Nothing found
- Update Anomaly: Nothing found
- Delete Anomaly: Nothing found

- 1NF : There is no multi-valued attribute in this relation. So this relation is already in 1NF form.
- 2NF : There is no partial dependency in this relation. So this relation is already in 2NF form.
- 3NF : There is no transitive dependency in this relation. So this relation is already in 3NF form.
- BCNF : This relation is already in BCNF form.

❖ team(**team_id**, t_name, no_players)

- Primary key : team_id
- Partial dependency: None
- Transitive dependency: None

- Insert Anomaly: Nothing found

- Update Anomaly: Nothing found
- Delete Anomaly: Nothing found

- 1NF: There is no multi-valued attribute in this relation. So this relation is already in 1NF form.
- 2NF: There is no partial dependency in this relation. So this relation is already in 2NF form.
- 3NF: There is no transitive dependency in this relation. So this relation is already in 3NF form.
- BCNF: This relation is already in BCNF form.

❖ manager(**manager_id**, role)

- Primary key : manager_id
- Foreign key : manager_id(person(id)), role(roles(r_name))
- Partial dependency : None
- Transitive dependency: None

- Insert Anomaly: Nothing found
- (Attributes and Reason)
- Update Anomaly: Nothing found
- Delete Anomaly: Nothing found

- 1NF : There is no multi-valued attribute in this relation. So this relation is already in 1NF form.
- 2NF : There is no partial dependency in this relation. So this relation is already in 2NF form.
- 3NF : There is no transitive dependency in this relation. So this relation is already in 3NF form.
- BCNF : This relation is already in BCNF form.

❖ worker(**worker_id**, role, speciality)

- Primary key : worker_id
- Foreign key : worker_id(person(id)), role(roles(r_name))
- Partial dependency : None
- Transitive dependency: None

- Insert Anomaly: Nothing found
- Update Anomaly: Nothing found
- Delete Anomaly: Nothing found

- 1NF : There is no multi-valued attribute in this relation. So this relation is already in 1NF form.
- 2NF : There is no partial dependency in this relation. So this relation is already in 2NF form.
- 3NF : There is no transitive dependency in this relation. So this relation is already in 3NF form.
- BCNF : This relation is already in BCNF form.

❖ game_history(**record_no**, game_id, player_id, date_and_time, rank, score, level, wins, losses, win_rate)

- Primary key : record_no
- Foreign key : game_id(game), player_id(player)
- Partial Dependencies:
 - game_id, player_id -> rank
 - game_id, player_id -> wins
 - game_id, player_id -> losses

- Transitive Dependencies:

- record_no-> wins,losses
wins,losses -> win_rate
record_no -> win_rate
- record_no, player_id,game_id -> score
score -> rank
record_no, player_id,game_id -> rank

- Insert Anomaly: For multiplayer games, if we insert any tuple in game_history then rank will be inconsistent as rank depends on score and number of players playing the game at that time, but at a time we can insert only one tuple. And without rank we can't calculate wins and losses and so winrate.
- Update Anomaly: If we change the score of any player then it reflects the change in rank of player in a particular game. And also if we change wins and losses, it also reflects a change in win_rate.
- Delete Anomaly: If we delete any tuple in game_history then we lose the information about the rank of player for a specific game and how many times the player wins or loses the game.
- 1NF : There is no multi-valued attribute in this relation. So this relation is already in 1NF form.
- 2NF : This relation is not in 2NF form. There is some partial dependency in this relation as mentioned above.
- 3NF : This relation is not in 3NF form. There is some transitive dependency in this relation as mentioned above.
- Updated schema:
 - player_history(player_id, table_no, date_and_time, score, win)
 - player_records(player_id, game_id, total_games, wins)

➤ BCNF : This relation is already in BCNF form.

❖ team_history(**record_no**, team_id, game_id, rank, date_and_time, score, level, wins, losses, win_rate)

➤ Primary key : record_no

➤ Foreign key : game_id(game), team_id(team)

➤ Partial Dependencies:

- game_id, team_id -> rank
- game_id, team_id -> wins
- game_id, team_id -> losses

➤ Transitive Dependencies:

- record_no-> wins,losses
wins,losses -> win_rate
record_no -> win_rate
- record_no, team_id, game_id -> score
score -> rank
record_no, team_id, game_id -> rank

➤ Insert Anomaly: For multiplayer games, if we insert any tuple in game_history then rank will be inconsistent as rank depends on score and number of teams playing the game at that time, but at a time we can insert only one tuple. And without rank we can't calculate wins and losses and so winrate.

➤ Update Anomaly: If we change the score of any team then it reflects the change in rank of team in a particular game. And also if we change wins and losses, it also reflects a change in win_rate.

➤ Delete Anomaly: If we delete any tuple in game_history then we lose the information about the rank of team for a specific game and how many times the team wins or loses the game.

- 1NF : There is no multi-valued attribute in this relation. So this relation is already in 1NF form.
- 2NF : This relation is not in 2NF form. There is some partial dependency in this relation as mentioned above.
- 3NF : This relation is not in 3NF form. There is some transitive dependency in this relation as mentioned above.
- Updated schema :
 - team_history(team id, table no, date and time, score, win)
 - team_records(team id, game id, total_games, wins)
- BCNF : This relation is already in BCNF form.

❖ lobby(**date and time, player id, game id**, no_of_players, status, login_time)

- Primary key : date_and_time, player_id, game_id
- Foreign key : game_id, player_id
- Partial Dependencies:
 - data_and_time -> login_time
- Transitive Dependencies: None
- Insert Anomaly: In a chase when two players are playing on the same table, then there are two tuples which have different values of no_of_players, which is inconsistent. So, this is the insert anomaly in this relation.
- Update Anomaly: if we update no_of_players, then it leads to inconsistency in status of players.
- Delete Anomaly: If we delete any tuple from this relation then it leads to the same problem in no_of_players and status values as we discussed above.

- 1NF : There is no multi-valued attribute in this relation. So this relation is already in 1NF form.
- 2NF : There is a partial dependency in this relation. So this relation is not in 2NF form.
We removed the login_time attribute from this relation. It removes partial dependency.
- 3NF : There is no transitive dependency in this relation. So this relation is already in 3NF form.
- Redundancy : Here we can measure no_of_player and login time from the value of other attributes.
- Update schema:
 - lobby(date and time, player id, table no, status)
- BCNF : This relation is already in BCNF form.

DDL Scripts:

1. Person

```
CREATE TABLE IF NOT EXISTS person (
  id int,
  first_name varchar(20) COLLATE pg_catalog."default" not null,
  middle_name varchar(20) COLLATE pg_catalog."default" not null,
  last_name varchar(20) COLLATE pg_catalog."default" not null,
  age int not null,
  contact_no char(10),
  email_address varchar(20) COLLATE pg_catalog."default" not NULL UNIQUE,
  user_type varchar(20) COLLATE pg_catalog."default" not null,
  username varchar(20) COLLATE pg_catalog."default" not null UNIQUE,
  password varchar(20) COLLATE pg_catalog."default" NOT NULL,
```

```
PRIMARY KEY (id),  
CHECK(user_type in ('manager','player','worker'))  
);
```

2. team

```
CREATE TABLE IF NOT EXISTS team (  
    team_id int,  
    t_name varchar(30) not null unique,  
    no_players int not null CHECK(no_players > 1),  
    PRIMARY KEY (team_id)  
);
```

3. Roles

```
CREATE TABLE IF NOT EXISTS Roles (  
    r_name varchar(20) COLLATE pg_catalog."default",  
    workhours int not null check(workhours > 0),  
    salary int not null check(salary > 0),  
    PRIMARY KEY (r_name)  
);
```

4. player

```
CREATE TABLE IF NOT EXISTS player (  
    player_id int,  
    height int not NULL,  
    weight int not NULL,  
    team_id int,  
    PRIMARY KEY (player_id),
```

```
FOREIGN KEY (player_id) REFERENCES person(id) MATCH SIMPLE ON UPDATE  
CASCADE ON DELETE CASCADE,  
FOREIGN KEY (team_id) REFERENCES team(team_id) MATCH SIMPLE ON  
UPDATE CASCADE ON DELETE CASCADE  
);
```

5. manager

```
CREATE TABLE IF NOT EXISTS manager (  
manager_id int,  
role varchar(20) COLLATE pg_catalog."default" NOT NULL,  
PRIMARY KEY (manager_id),  
FOREIGN KEY (role) REFERENCES roles(r_name) MATCH SIMPLE ON UPDATE  
CASCADE ON DELETE CASCADE,  
FOREIGN KEY (manager_id) REFERENCES person(id) MATCH SIMPLE ON  
UPDATE CASCADE ON DELETE CASCADE  
);
```

6. worker

```
CREATE TABLE IF NOT EXISTS worker (  
worker_id int,  
role varchar(20) COLLATE pg_catalog."default" not NULL,  
PRIMARY KEY (worker_id),  
FOREIGN KEY (role) REFERENCES roles(r_name) MATCH SIMPLE ON UPDATE  
CASCADE ON DELETE CASCADE,  
FOREIGN KEY (worker_id) REFERENCES person(id) MATCH SIMPLE ON UPDATE  
CASCADE ON DELETE CASCADE  
);
```

7. account

```
CREATE TABLE IF NOT EXISTS account (  
    account_no char(11),  
    player_id int not NULL UNIQUE,  
    debit_card_no char(16) not NULL ,  
    date_of_joining date not NULL,  
    balance int default 1000,  
    PRIMARY KEY (account_no),  
    FOREIGN KEY (player_id) REFERENCES player(player_id) MATCH SIMPLE ON  
UPDATE CASCADE ON DELETE CASCADE  
);
```

8. transactions

```
CREATE TABLE IF NOT EXISTS transactions (  
    transaction_id char(15),  
    account_no char(11) not NULL,  
    date_and_time timestamp not NULL,  
    amount int not NULL,  
    type varchar(20) COLLATE pg_catalog."default" not NULL,  
    method varchar(15),  
    PRIMARY KEY (transaction_id),  
    FOREIGN KEY (account_no) REFERENCES account(account_no) MATCH  
SIMPLE ON UPDATE RESTRICT ON DELETE RESTRICT  
);
```

9. game

```
CREATE TABLE IF NOT EXISTS game (  
  game_id int,  
  game_name varchar(20) COLLATE pg_catalog."default" not null unique,  
  type varchar(20) COLLATE pg_catalog."default" ,  
  mode varchar(20) COLLATE pg_catalog."default" not null,  
  price int not null CHECK(price > 0),  
  reward int default 0,  
  age int default 5,  
  height int,  
  weight int,  
  PRIMARY KEY (game_id),  
  CHECK(mode in ('online','offline'))  
);
```

10.purchase

```
CREATE TABLE IF NOT EXISTS purchase (  
  purchase_id int,  
  player_id int,  
  game_id int ,  
  no_hours int not null CHECK(no_hours > 0),  
  PRIMARY KEY (purchase_id),  
  FOREIGN KEY (player_id) REFERENCES player(player_id) MATCH SIMPLE ON  
UPDATE CASCADE ON DELETE CASCADE,  
  FOREIGN KEY (game_id) REFERENCES game(game_id) MATCH SIMPLE ON  
UPDATE CASCADE ON DELETE CASCADE  
);
```


11. game_table

```
CREATE TABLE IF NOT EXISTS game_table (  
    table_no int,  
    game_id int,  
    capacity int not null check(capacity>0),  
    PRIMARY KEY (table_no),  
    FOREIGN KEY (game_id) REFERENCES game(game_id) MATCH SIMPLE ON  
UPDATE CASCADE ON DELETE CASCADE  
);
```

12. player_history

```
CREATE TABLE IF NOT EXISTS player_history (  
    player_id int,  
    table_no int,  
    date_and_time timestamp not NULL,  
    score int not NULL CHECK(score >=0),  
    win boolean default false,  
    PRIMARY KEY (player_id,table_no),  
    FOREIGN KEY (table_no) REFERENCES game_table(table_no) MATCH SIMPLE  
ON UPDATE CASCADE ON DELETE CASCADE,  
    FOREIGN KEY (player_id) REFERENCES player(player_id) MATCH SIMPLE ON  
UPDATE CASCADE ON DELETE CASCADE  
);
```

13. player_records

```
CREATE TABLE IF NOT EXISTS player_records (  
    player_id int,  
    game_id int,
```

```
total_games int default 0,  
wins int default 0 check(wins<=total_games),  
PRIMARY KEY (player_id,game_id),  
FOREIGN KEY (game_id) REFERENCES game(game_id) MATCH SIMPLE ON  
UPDATE CASCADE ON DELETE CASCADE,  
FOREIGN KEY (player_id) REFERENCES player(player_id) MATCH SIMPLE ON  
UPDATE CASCADE ON DELETE CASCADE  
);
```

14. team_history

```
CREATE TABLE IF NOT EXISTS team_history (  
team_id int,  
table_no int,  
date_and_time timestamp not NULL,  
score int not NULL CHECK(score >=0),  
win boolean default false,  
PRIMARY KEY (team_id,table_no),  
FOREIGN KEY (table_no) REFERENCES game_table(table_no) MATCH SIMPLE  
ON UPDATE CASCADE ON DELETE CASCADE,  
FOREIGN KEY (team_id) REFERENCES team(team_id) MATCH SIMPLE ON  
UPDATE CASCADE ON DELETE CASCADE  
);
```

15. team_records

```
CREATE TABLE IF NOT EXISTS team_records (  
team_id int,  
game_id int,  
total_games int default 0,  
wins int default 0 check(wins<=total_games),
```

```
PRIMARY KEY (team_id,game_id),
FOREIGN KEY (game_id) REFERENCES game(game_id) MATCH SIMPLE ON
UPDATE CASCADE ON DELETE CASCADE,
FOREIGN KEY (team_id) REFERENCES team(team_id) MATCH SIMPLE ON
UPDATE CASCADE ON DELETE CASCADE
);
```

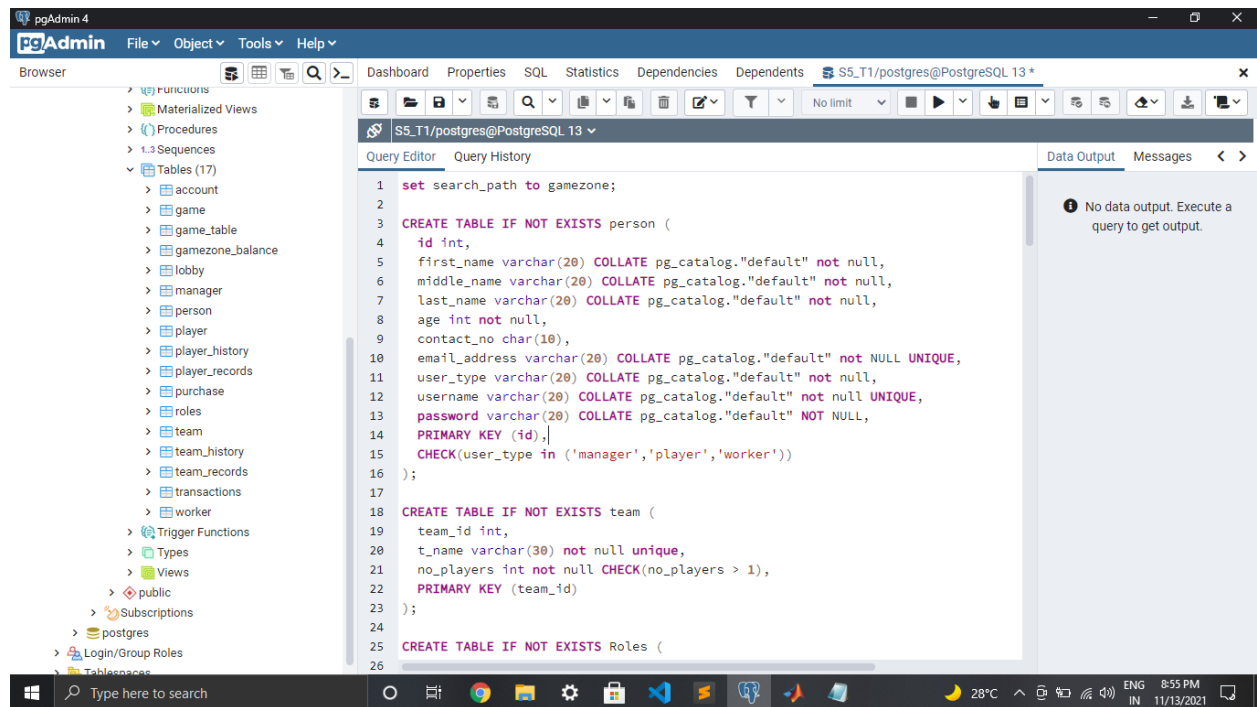
16. lobby

```
CREATE TABLE IF NOT EXISTS lobby (
  player_id int,
  table_no int,
  date_and_time timestamp,
  status varchar(20) COLLATE pg_catalog."default",
  PRIMARY KEY (player_id, table_no, date_and_time),
  FOREIGN KEY (player_id) REFERENCES player(player_id) MATCH SIMPLE ON
  UPDATE CASCADE ON DELETE CASCADE,
  FOREIGN KEY (table_no) REFERENCES game_table(table_no) MATCH SIMPLE
  ON UPDATE CASCADE ON DELETE CASCADE,
  CHECK (status in ('playing','waiting'))
);
```

17. GameZone_balance

```
CREATE TABLE IF NOT EXISTS GameZone_balance (
  account_no char(11) NOT NULL,
  balance int default 50000
);
```

DDL SnapShot:

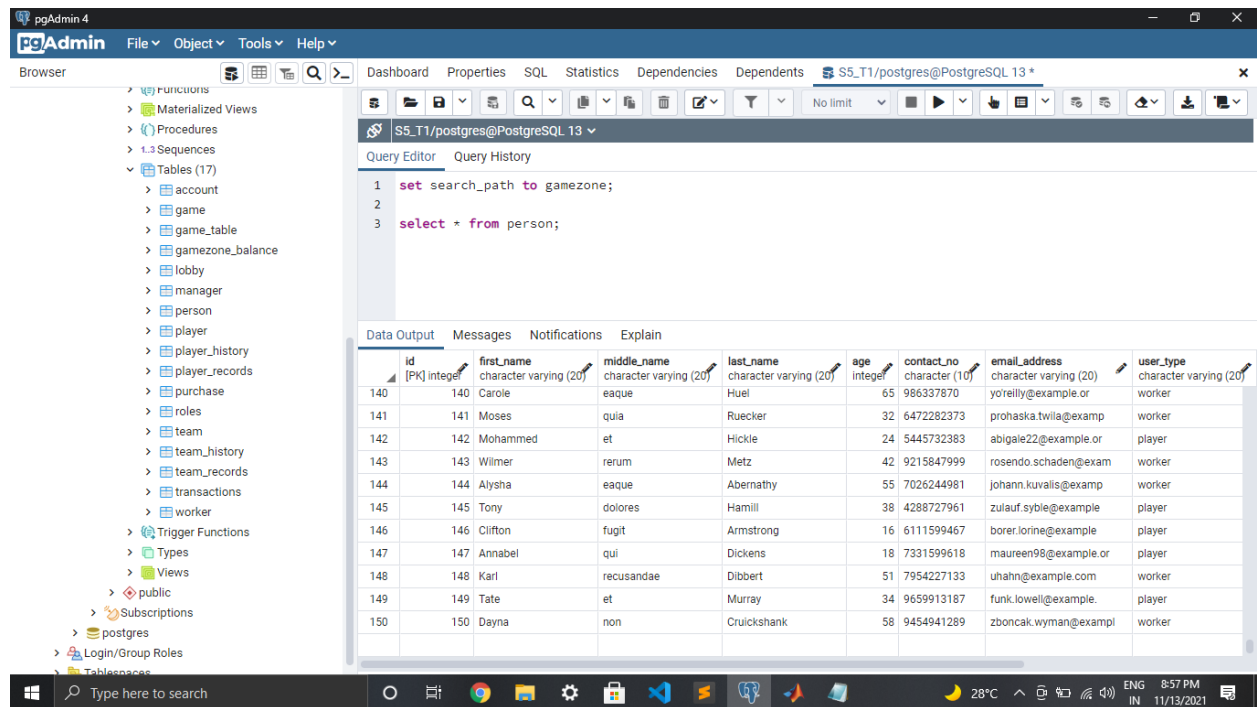


The screenshot shows the pgAdmin 4 interface with the Query Editor open. The query contains DDL statements for creating tables and roles.

```
1 set search_path to gamezone;
2
3 CREATE TABLE IF NOT EXISTS person (
4     id int,
5     first_name varchar(20) COLLATE pg_catalog."default" not null,
6     middle_name varchar(20) COLLATE pg_catalog."default" not null,
7     last_name varchar(20) COLLATE pg_catalog."default" not null,
8     age int not null,
9     contact_no char(10),
10    email_address varchar(20) COLLATE pg_catalog."default" not null UNIQUE,
11    user_type varchar(20) COLLATE pg_catalog."default" not null,
12    username varchar(20) COLLATE pg_catalog."default" not null UNIQUE,
13    password varchar(20) COLLATE pg_catalog."default" not null,
14    PRIMARY KEY (id),
15    CHECK(user_type in ('manager','player','worker'))
16 );
17
18 CREATE TABLE IF NOT EXISTS team (
19     team_id int,
20     t_name varchar(30) not null unique,
21     no_players int not null CHECK(no_players > 1),
22     PRIMARY KEY (team_id)
23 );
24
25 CREATE TABLE IF NOT EXISTS Roles (
26
```

The right pane shows a message: "No data output. Execute a query to get output."

person:



The screenshot shows the pgAdmin 4 interface with the Query Editor open. The query is a SELECT statement for the 'person' table.

```
1 set search_path to gamezone;
2
3 select * from person;
```

The right pane shows the Data Output tab with the following data:

	id	first_name	middle_name	last_name	age	contact_no	email_address	user_type
140	140	Carole	eaque	Huel	65	986337870	yo'reilly@example.or	worker
141	141	Moses	quia	Ruecker	32	6472282373	prohaska.twila@examp	worker
142	142	Mohammed	et	Hickle	24	5445732383	abigale22@example.or	player
143	143	Wilmer	rerum	Metz	42	9215847999	rosendo.schaden@exam	worker
144	144	Alysha	eaque	Abernathy	55	7026244981	johann.kuvalis@examp	worker
145	145	Tony	dolores	Hamill	38	4288727961	zulauf.syble@example	player
146	146	Clifton	fugit	Armstrong	16	6111599467	borer.lorine@example	player
147	147	Annabel	qui	Dickens	18	7331599618	maureen98@example.or	player
148	148	Kari	recusandae	Dibbert	51	7954227133	uhahn@example.com	worker
149	149	Tate	et	Murray	34	9659913187	funk.lowell@example.	player
150	150	Dayna	non	Cruickshank	58	9454941289	zboncak.wyman@examp	worker

No. of Tuples = 150

team:

The screenshot shows the pgAdmin 4 interface. The left sidebar displays a tree view of the database schema, including tables like 'team'. The main window is the 'Query Editor' for the 'S5_T1/postgres@PostgreSQL 13' database. The query entered is:

```
1 set search_path to gamezone;  
2  
3 select * from team;
```

The 'Data Output' tab is active, showing the results of the query. The table has three columns: 'team_id' (PK Integer), 't_name' (character varying (30)), and 'no_players' (Integer). The results are as follows:

team_id	t_name	no_players
15	et	3
16	autem	3
17	dolor	3
18	vel	2
19	maiores	2
20	molestiae	2
21	sunt	3
22	ipsa	2
23	temporibus	2
24	aliquam	2
25	est	4

No of Tuples: 25

roles:

The screenshot shows the pgAdmin 4 interface. The left sidebar displays a tree view of the database schema, including tables like 'roles'. The main window is the 'Query Editor' for the 'S5_T1/postgres@PostgreSQL 13' database. The query entered is:

```
1 set search_path to gamezone;  
2  
3 select * from roles;
```

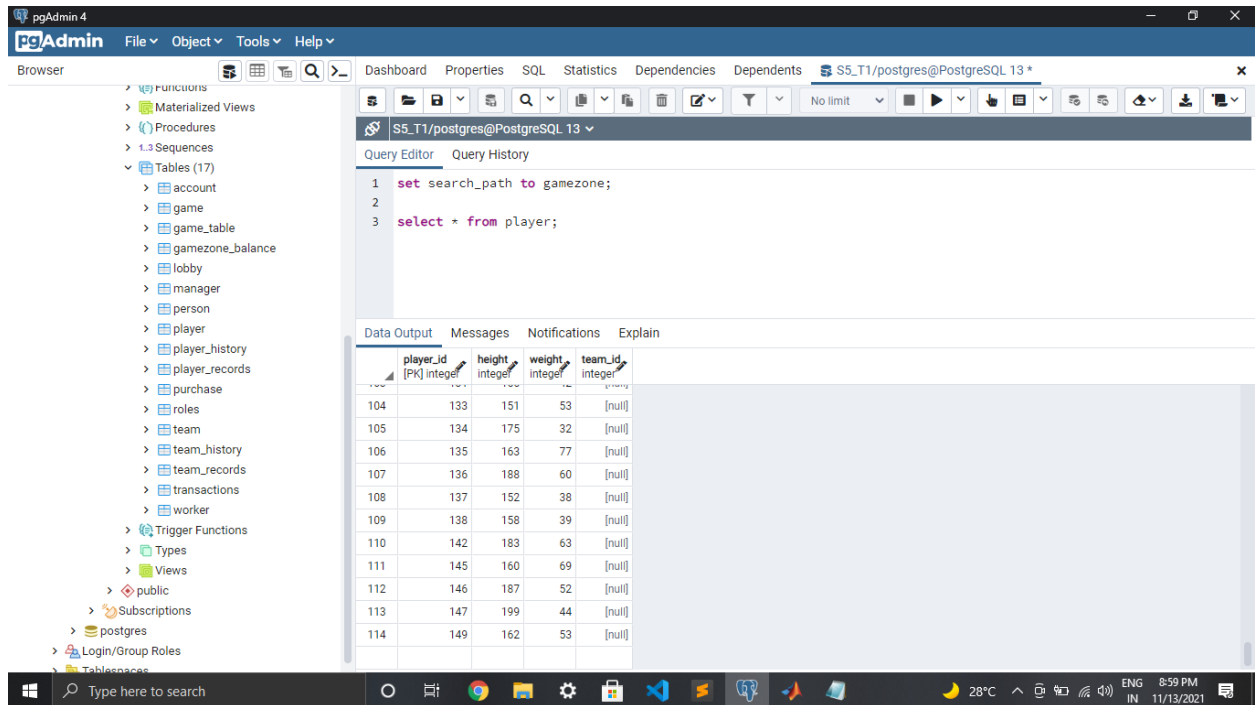
The 'Data Output' tab is active, showing the results of the query. The table has three columns: 'r_name' (PK character varying (20)), 'workhours' (Integer), and 'salary' (Integer). The results are as follows:

r_name	workhours	salary
5 ab	10	15818
6 est	13	46080
7 fuga	11	29732
8 ut	12	49954
9 hic	10	16748
10 vitae	12	44585
11 exercitationem	10	21001
12 ipsa	11	49750
13 laboriosam	14	19555
14 porro	12	16001
15 quia	12	39561

A green status bar at the bottom right indicates: 'Successfully run. Total query runtime: 96 ms'.

No of Tuples: 15

player:



The screenshot shows the pgAdmin 4 interface. The left sidebar displays a tree view of the database structure, including tables like 'player', 'team', and 'game'. The main window is titled 'S5_T1/postgres@PostgreSQL 13 *'. The 'Query Editor' tab is active, showing the following SQL query:

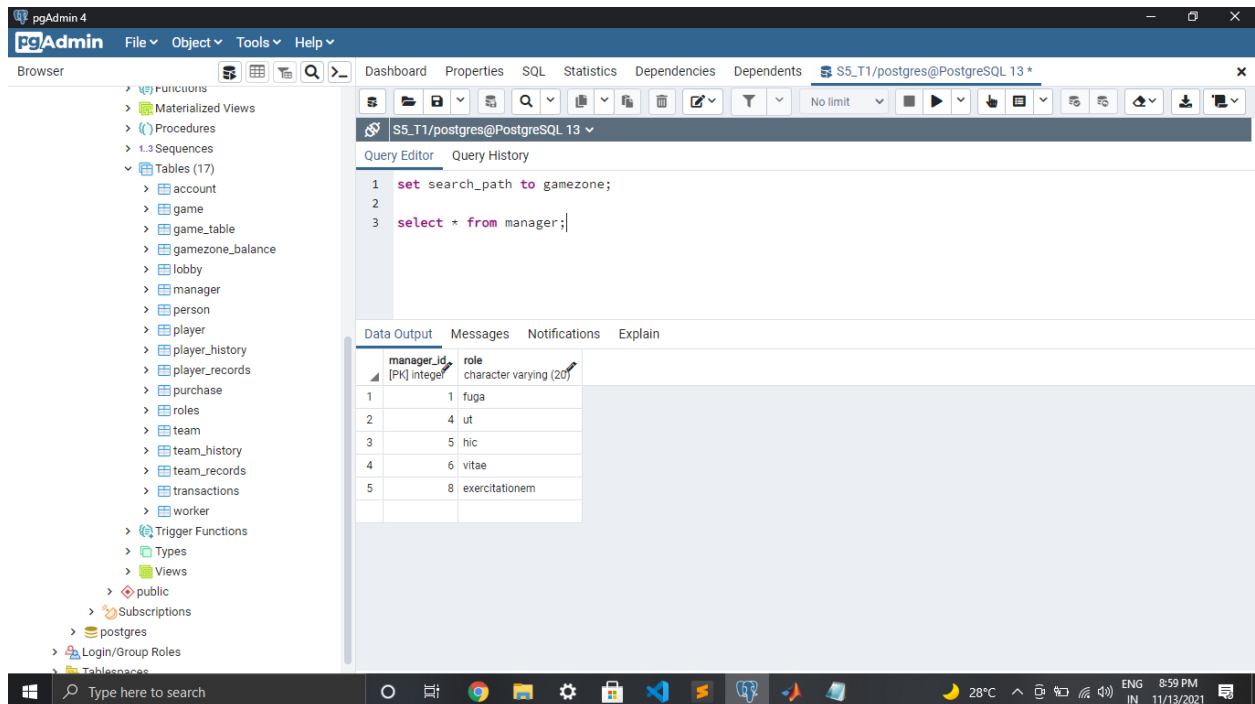
```
1 set search_path to gamezone;  
2  
3 select * from player;
```

The 'Data Output' tab is selected, displaying the results of the query in a table format. The table has four columns: 'player_id' (PK integer), 'height' (integer), 'weight' (integer), and 'team_id' (integer). There are 114 rows of data.

player_id	height	weight	team_id
104	133	151	53
105	134	175	32
106	135	163	77
107	136	188	60
108	137	152	38
109	138	158	39
110	142	183	63
111	145	160	69
112	146	187	52
113	147	199	44
114	149	162	53

No of Tuples: 114

manager:



The screenshot shows the pgAdmin 4 interface. The left sidebar displays a tree view of the database structure, including tables like 'manager', 'team', and 'game'. The main window is titled 'S5_T1/postgres@PostgreSQL 13 *'. The 'Query Editor' tab is active, showing the following SQL query:

```
1 set search_path to gamezone;  
2  
3 select * from manager;
```

The 'Data Output' tab is selected, displaying the results of the query in a table format. The table has two columns: 'manager_id' (PK integer) and 'role' (character varying (20)). There are 5 rows of data.

manager_id	role
1	fuga
2	ut
3	hic
4	vitae
5	exercitationem

No of Tuples: 5

worker:

The screenshot shows the pgAdmin 4 interface. The left sidebar displays a tree view of the database structure, with 'Tables (17)' expanded. The 'worker' table is selected. The main pane shows the 'Query Editor' with the following SQL query:

```
1 set search_path to gamezone;  
2  
3 select * from worker;
```

The 'Data Output' tab is active, displaying the results of the query in a table format. The table has two columns: 'worker_id' (integer) and 'role' (character varying (20)). The results are as follows:

worker_id	role
21	125 nobis
22	126 ab
23	127 est
24	132 quia
25	139 voluptate
26	140 nobis
27	141 ab
28	143 est
29	144 quia
30	148 porro
31	150 quia

No of Tuples: 31

account:

The screenshot shows the pgAdmin 4 interface. The left sidebar displays a tree view of the database structure, with 'Tables (17)' expanded. The 'account' table is selected. The main pane shows the 'Query Editor' with the following SQL query:

```
1 set search_path to gamezone;  
2  
3 select * from account;
```

The 'Data Output' tab is active, displaying the results of the query in a table format. The table has five columns: 'account_no' (integer), 'player_id' (integer), 'debit_card_no' (integer), 'date_of_joining' (date), and 'balance' (integer). The results are as follows:

account_no	player_id	debit_card_no	date_of_joining	balance	
104	90700643421	133	8.3272E+15	2003-03-19	72255
105	90913211880	134	8.64855E+15	1992-08-19	49007
106	91495038713	135	7.17944E+15	2003-02-04	47864
107	91767886132	136	8.48857E+15	1972-07-06	62029
108	93472974575	137	5.07496E+15	2000-06-18	79963
109	93911828184	138	5.99733E+15	2006-07-16	47944
110	95218122648	142	9.78808E+15	2006-08-30	77614
111	95563080543	145	3.27724E+15	1991-09-11	69706
112	96879522013	146	1.78856E+14	1977-02-23	35113
113	97112491647	147	1.93289E+15	1989-01-05	44342
114	98909540255	149	3.03998E+15	2007-09-03	35161

No of Tuples: 114

transaction:

The screenshot shows the pgAdmin 4 interface with a query executed on the 'transactions' table. The query is:

```
1 set search_path to gamezone;  
2  
3 select * from transactions;
```

The result set contains 11 columns and 11 rows of data:

	transaction_id [PK] character (15)	account_no character (11)	date_and_time timestamp without time zone	amount integer	type character varying (20)	method character varying (15)
490	98576wzjk55gh82	31949669425	1993-05-03 13:45:00	2624	loss	Paytm
491	98589noys00pm39	33442892273	2016-10-19 18:45:00	3651	pay	Check
492	98605xvib38cb08	34092686283	1996-11-26 19:43:00	1746	loss	Check
493	98640nqgf02id12	34462143266	1982-08-25 19:05:00	3352	pay	Cash
494	98751mij49if79	34539477429	1988-07-02 20:23:00	4356	pay	Gpay
495	98999dyf89fm52	35477835889	1984-11-15 21:06:00	1323	pay	Paytm
496	99132yprw69ep96	36613480085	1982-11-24 23:18:00	4944	pay	Paytm
497	99543yily11nj42	38392742085	1976-05-10 05:14:00	997	win	Gpay
498	99615sqcp36aa27	38453895193	1989-05-24 10:40:00	2720	pay	Paytm
499	99924ffsc28wf33	38864569137	2008-07-09 04:01:00	4822	loss	Check
500	99981ytiv75bd85	39105557571	2021-11-05 07:11:00	4370	win	Paytm

No of Tuples: 500

game:

The screenshot shows the pgAdmin 4 interface with a query executed on the 'game' table. The query is:

```
1 set search_path to gamezone;  
2  
3 select * from game;
```

The result set contains 11 columns and 11 rows of data:

	game_id [PK] integer	game_name character varying (20)	type character varying (20)	mode character varying (20)	price integer	reward integer	age integer	height integer	weight integer
30	30	quo	Sports	offline	60630	586	67	165	83
31	31	deleniti	Sports	offline	34702	962	39	107	47
32	32	praesentium	Action	offline	57266	1000	39	191	44
33	33	et	Adventure	offline	74332	903	6	200	20
34	34	quasi	Board	offline	12369	952	61	152	25
35	35	perferendis	Board	online	55065	966	8	130	51
36	36	consequatur	Racing	online	18211	928	45	166	63
37	37	in	Casino	offline	3730	898	53	162	61
38	38	quidem	Board	online	14569	928	11	167	62
39	39	temporibus	Sports	online	30750	745	55	163	73
40	40	rerum	Racing	offline					

Successfully run. Total query runtime: 85 msec. 40 rows affected.

No of Tuples: 40

game_table:

The screenshot shows the pgAdmin 4 interface with the 'game_table' selected in the browser. The query editor contains the following SQL:

```
1 set search_path to gamezone;  
2  
3 select * from game_table;
```

The 'Data Output' tab displays the results of the query. The table has 150 rows and 4 columns: table_no (PK integer), game_id integer, capacity integer, and an unnamed column. The data is as follows:

table_no (PK integer)	game_id integer	capacity integer	
140	140	30	3
141	141	9	2
142	142	27	5
143	143	27	3
144	144	20	3
145	145	8	2
146	146	28	6
147	147	29	6
148	148	40	5
149	149	40	6
150	150	13	5

No of Tuples: 150

player_history:

The screenshot shows the pgAdmin 4 interface with the 'player_history' selected in the browser. The query editor contains the following SQL:

```
1 set search_path to gamezone;  
2  
3 select * from player_history;
```

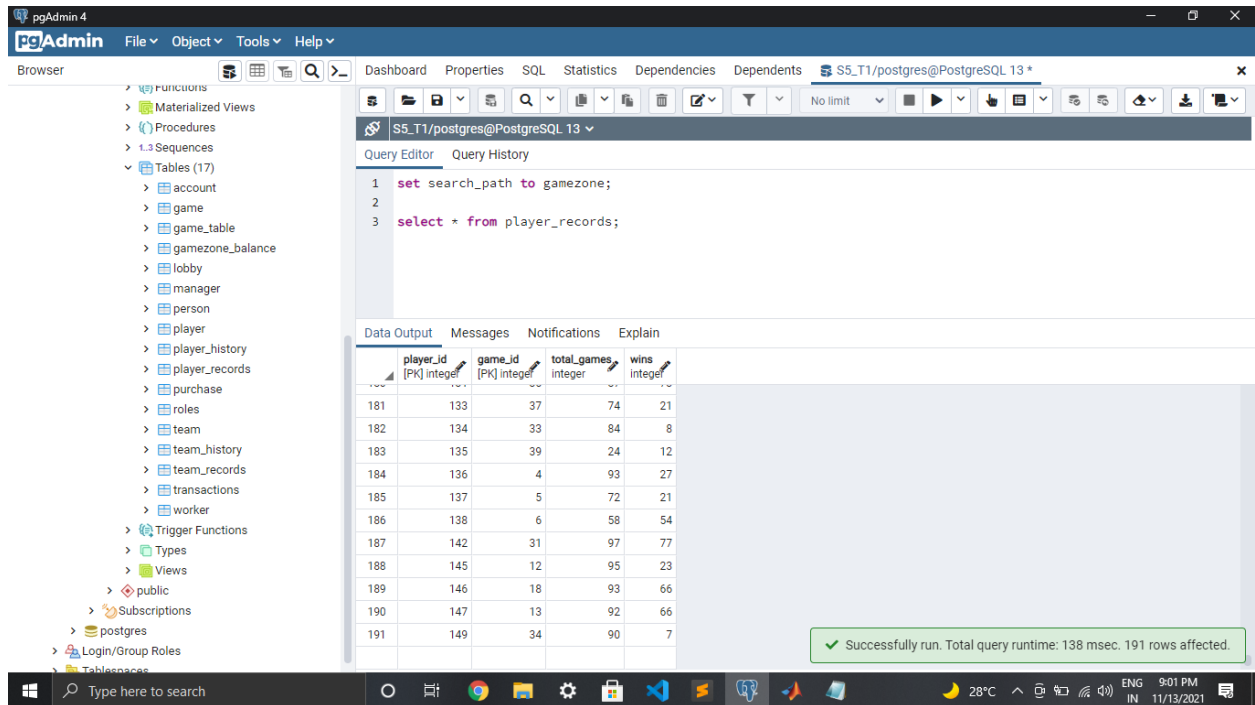
The 'Data Output' tab displays the results of the query. The table has 200 rows and 6 columns: player_id (PK integer), table_no (PK integer), date_and_time timestamp without time zone, score integer, win boolean, and an unnamed column. The data is as follows:

player_id (PK integer)	table_no (PK integer)	date_and_time timestamp without time zone	score integer	win boolean	
190	133	83 1987-01-20 10:03:00	3169	true	
191	134	145 2001-01-15 03:11:00	4954	false	
192	135	12 1992-12-25 21:15:00	4214	true	
193	136	96 1979-09-24 03:38:00	3235	true	
194	137	136 1970-12-30 00:59:00	2689	true	
195	138	19 1997-03-11 03:14:00	2260	true	
196	142	87 2016-04-20 07:45:00	4173	false	
197	145	109 1981-08-07 01:16:00	2585	true	
198	146	3 1980-02-14 11:00:00	381	true	
199	147	39 2012-04-19 13:44:00	499	false	
200	149	107 2016-05-31 01:29:00	4838	false	

Successfully run. Total query runtime: 100 msec. 200 rows affected.

No of Tuples: 200

player_records:



The screenshot shows the pgAdmin 4 interface with a query executed on the `player_records` table. The query editor contains the following SQL:

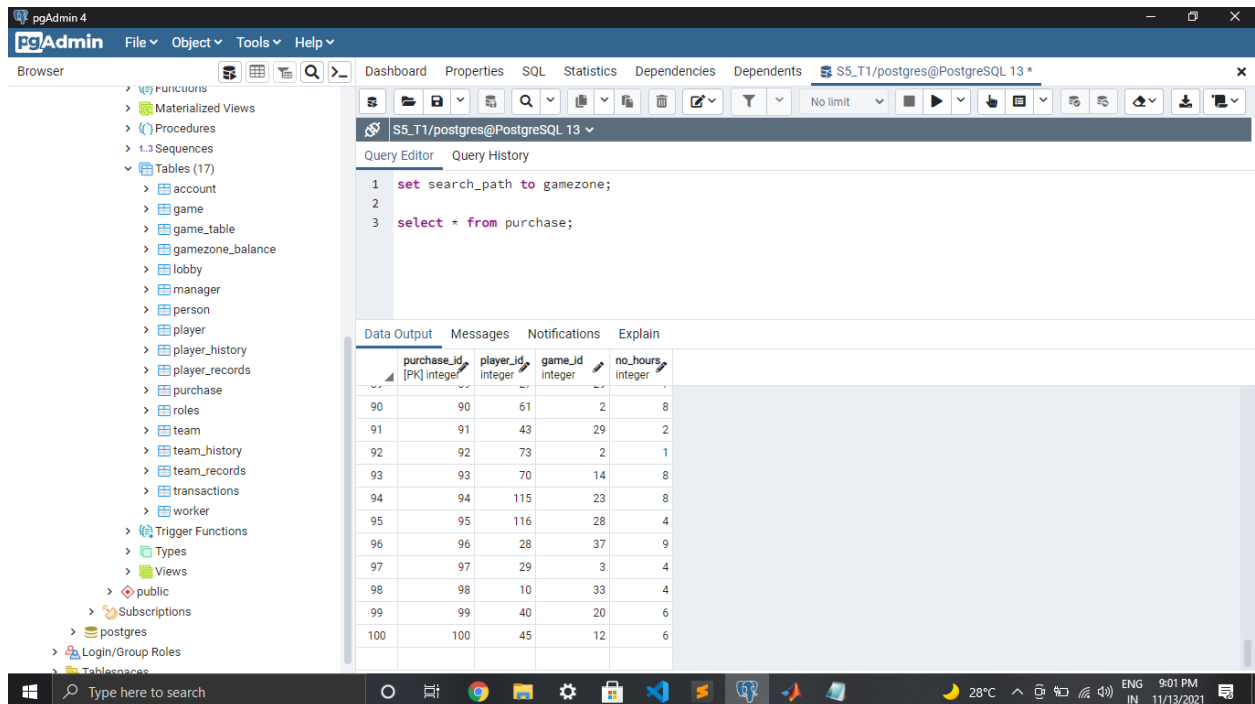
```
1 set search_path to gamezone;  
2  
3 select * from player_records;
```

The Data Output tab displays the results of the query, showing 191 rows. The columns are `player_id` (PK integer), `game_id` (PK integer), `total_games` integer, and `wins` integer. The status bar at the bottom indicates: "Successfully run. Total query runtime: 138 msec. 191 rows affected."

player_id	game_id	total_games	wins
181	133	37	74
182	134	33	84
183	135	39	24
184	136	4	93
185	137	5	72
186	138	6	58
187	142	31	97
188	145	12	95
189	146	18	93
190	147	13	92
191	149	34	90

No of Tuples: 191

purchase:



The screenshot shows the pgAdmin 4 interface with a query executed on the `purchase` table. The query editor contains the following SQL:

```
1 set search_path to gamezone;  
2  
3 select * from purchase;
```

The Data Output tab displays the results of the query, showing 100 rows. The columns are `purchase_id` (PK integer), `player_id` integer, `game_id` integer, and `no_hours` integer. The status bar at the bottom indicates: "Successfully run. Total query runtime: 138 msec. 191 rows affected."

purchase_id	player_id	game_id	no_hours
90	90	61	2
91	91	43	29
92	92	73	2
93	93	70	14
94	94	115	23
95	95	116	28
96	96	28	37
97	97	29	3
98	98	10	33
99	99	40	20
100	100	45	12

No of Tuples: 100

team_history:

The screenshot shows the pgAdmin 4 interface with the 'team_history' table selected in the 'Tables (17)' folder. The 'Query Editor' tab is active, displaying the following SQL query:

```
1 set search_path to gamezone;  
2  
3 select * from team_history;
```

The 'Data Output' tab shows the results of the query, which are 147 rows. The table has the following columns: team_id (PK integer), table_no (PK integer), date_and_time (timestamp without time zone), score (integer), and win (boolean). The data is as follows:

team_id	table_no	date_and_time	score	win
137	23	2010-05-31 12:43:00	4376	true
138	23	2017-03-05 12:42:00	2243	true
139	23	2007-04-19 23:02:00	3463	false
140	24	2005-11-24 22:41:00	3927	false
141	24	1972-10-14 12:39:00	1115	true
142	24	1995-06-13 20:23:00	1237	true
143	25	1973-11-27 06:48:00	4564	false
144	25	2003-12-26 10:40:00	2784	false
145	25	1983-06-17 09:12:00	3547	false
146	25	2020-08-20 17:54:00	2261	false
147	25	1992-03-07 18:48:00	4406	false

A green status bar at the bottom of the data output area indicates: 'Successfully run. Total query runtime: 163 msec. 147 rows affected.'

No of Tuples: 147

team_records:

The screenshot shows the pgAdmin 4 interface with the 'team_records' table selected in the 'Tables (17)' folder. The 'Query Editor' tab is active, displaying the following SQL query:

```
1 set search_path to gamezone;  
2  
3 select * from team_records;
```

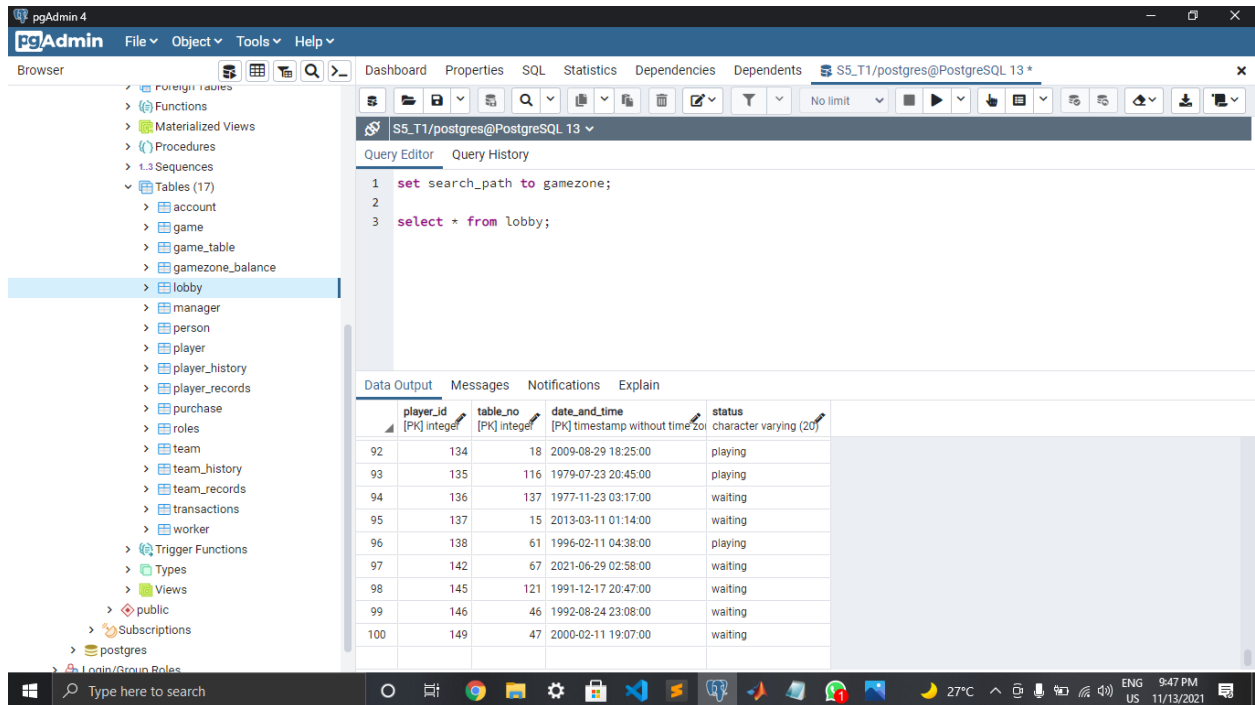
The 'Data Output' tab shows the results of the query, which are 145 rows. The table has the following columns: team_id (PK integer), game_id (PK integer), total_games (integer), and wins (integer). The data is as follows:

team_id	game_id	total_games	wins
135	23	38	58
136	24	10	72
137	24	22	38
138	24	34	65
139	25	5	36
140	25	13	52
141	25	18	92
142	25	22	50
143	25	30	86
144	25	33	63
145	25	35	65

A green status bar at the bottom of the data output area indicates: 'Successfully run. Total query runtime: 101 msec. 145 rows affected.'

No of Tuples: 145

lobby:



The screenshot shows the pgAdmin 4 interface. The left sidebar displays a tree view of the database schema, with the 'lobby' table selected under the 'Tables' category. The main window shows the 'Query Editor' with the following SQL query:

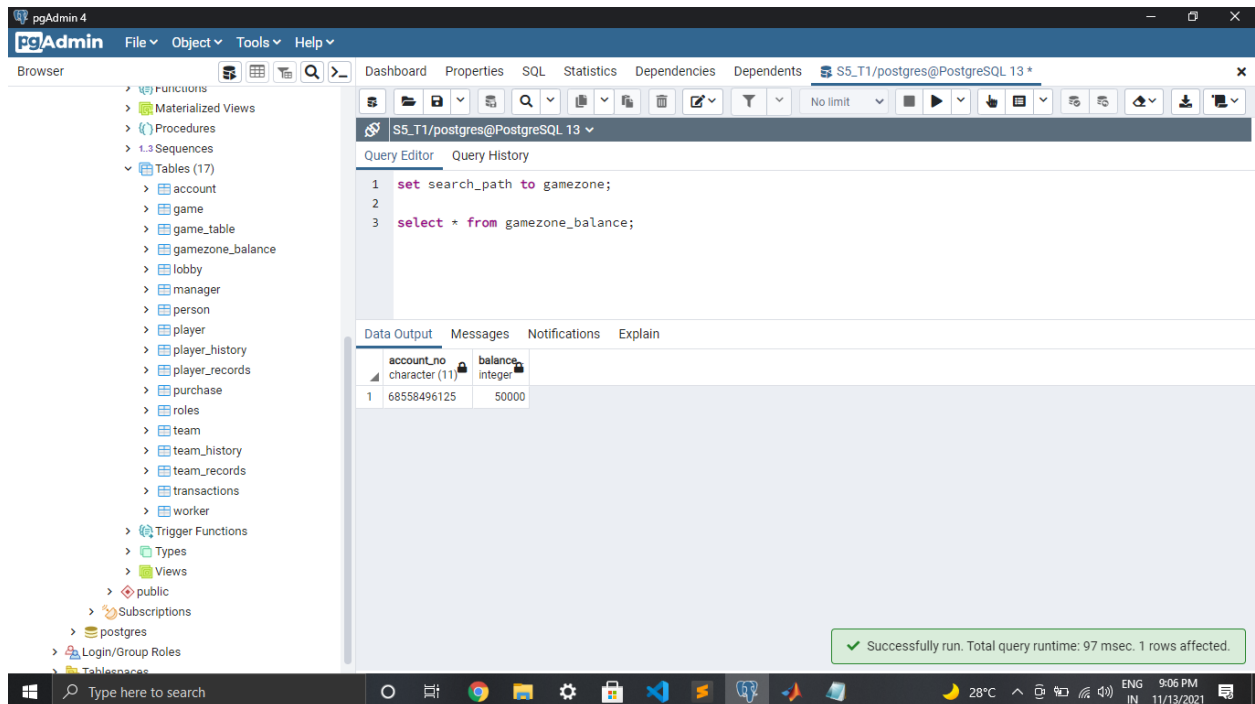
```
1 set search_path to gamezone;  
2  
3 select * from lobby;
```

The 'Data Output' tab is active, displaying the results of the query in a table format. The table has five columns: 'player_id' (integer), 'table_no' (integer), 'date_and_time' (timestamp without time zone), and 'status' (character varying (20)). There are 100 rows of data.

player_id	table_no	date_and_time	status
92	134	2009-08-29 18:25:00	playing
93	135	1979-07-23 20:45:00	playing
94	136	1977-11-23 03:17:00	waiting
95	137	2013-03-11 01:14:00	waiting
96	138	1996-02-11 04:38:00	playing
97	142	2021-06-29 02:58:00	waiting
98	145	1991-12-17 20:47:00	waiting
99	146	1992-08-24 23:08:00	waiting
100	149	2000-02-11 19:07:00	waiting

No of Tuples: 100

gamezone_balance:



The screenshot shows the pgAdmin 4 interface. The left sidebar displays a tree view of the database schema, with the 'gamezone_balance' table selected under the 'Tables' category. The main window shows the 'Query Editor' with the following SQL query:

```
1 set search_path to gamezone;  
2  
3 select * from gamezone_balance;
```

The 'Data Output' tab is active, displaying the results of the query in a table format. The table has two columns: 'account_no' (character (11)) and 'balance' (integer). There is 1 row of data.

account_no	balance
68558496125	50000

A green status bar at the bottom of the query editor indicates: "Successfully run. Total query runtime: 97 msec. 1 rows affected."

No of Tuples: 1