**THE MINISTRY OF EDUCATION AND SCIENCE OF THE REPUBLIC OF KAZAKHSTAN**

**INTERNATIONAL INFORMATION TECHNOLOGIES UNIVERSITY**

**Faculty of Information technologies**

**Department of Computer Engineering and Telecommunications**

**COURSE WORK**

**Topic**

|  |  |
| --- | --- |
| **Student: Maksatkaliuly Adilet,Aksakalov Beksultan**  **Group: CSSE-1501**  **Accepted :** |  |

**Almaty 2016**

**Contents**

Subject Area Analysis------------------------------------------------------------------------------------------- 4

**Introduction**

## Course work. Part 3

**Logical Design (DB of individual topic)**

**Goal:** to translate the conceptual data model into a logical data model and then to validate this model to check that it is structurally correct. A logical data model includes the complete ER-diagram and a supporting information.

C:\Users\Бексултан\Downloads\Aksakalov_Beksultan_and_Adilet_Maksatkaliuly_course work.Part2_CSSE1501K.png

ER-diagram :

PLAN:

Complete ER-diagram to the 3NF (third normal form). Describe all your actions to achieve it.

During the creating ER- diagram we tied two entities a many-to-many. For example: the roles and users. It was our fault, because the one user can have several roles and one role can belong to multiple users. After that we have added third “User\_roles” entity .

According to the 3NF our tables depends from PK and we use many FK to relate with other tables

Describe every entity of the database like in the following example. For the column with attribute’s types use: N – for numeric, C – for character, D – for date.

Entity users

|  |  |  |  |
| --- | --- | --- | --- |
| Content | Name | Type,length | note |
| ID of user | Id | Integer(11) unsigned | PK |
| First name | first\_name | Varchar(255) | Mandatory field |
| Last name | last\_name | Varchar(255) | Mandatory field |
| Middle name | middle\_name | Varchar(255) |  |
| Individual number | IIN | Varchar(255) | Mandatory field |
| doc\_type | doc\_type | Varchar(20) | Mandatory field |
| doc\_number | doc\_number | Varchar(50) | Mandatory field |
| doc\_issue | doc\_issue | Varchar(50) | Mandatory field |
| doc\_expire | doc\_expire | Varchar(50) | Mandatory field |

Entity roles

|  |  |  |  |
| --- | --- | --- | --- |
| Content | Name | Type,length | note |
| Id of role | id | Integer(11) unsigned | PK |
| Description of role | description | Varchar(255) | Mandatory field |

Entity orders

|  |  |  |  |
| --- | --- | --- | --- |
| Content | Name | Type,length | note |
|  |  |  |  |
| Id of order | id | Integer(11) unsigned | PK |
| order\_time | order\_time | Varchar(50) | Mandatory field |
| user\_id | user\_id | Integer(11) unsigned | FK to users |
| ticket\_id | ticket\_id | Integer(11) unsigned | FK to tickets |

Entity tickets

|  |  |  |  |
| --- | --- | --- | --- |
| Content | Name | Type,length | note |
| Id of ticket | id | Integer(11) unsigned | PK |
|  |  |  |  |
| print\_time | print\_time | Varchar(50) | Mandatory field |
| place\_time | place\_time | Varchar(50) | Mandatory field |

Entity Places

|  |  |  |  |
| --- | --- | --- | --- |
| Content | Name | Type,length | note |
| Id of place | id | Integer(11) unsigned | PK |
|  |  |  |  |
| price | price | Integer(11) unsigned | Mandatory field |
| train\_id | train\_id | Integer(11) unsigned | FK to trains |
| classification\_type\_id | classification\_type\_id | Integer(11) unsigned | FK to classification types |

Entity classification\_types

|  |  |  |  |
| --- | --- | --- | --- |
| Content | Name | Type,length | note |
| Id of classification\_type | id | Integer(11) unsigned | PK |
| name | name | Varchar(255) | Mandatory field |
| description | description | Varchar(255) | Mandatory field |

Table compare\_places

|  |  |  |  |
| --- | --- | --- | --- |
| Content | Name | Type,length | note |
|  |  |  |  |
| Id of compare\_places | id | Integer(11) unsigned | PK |
| user\_id | user\_id | Integer(11) unsigned | FK to users |
| place\_id | place\_id | Integer(11) unsigned | FK to placesFK to users |

Table Trains

|  |  |  |  |
| --- | --- | --- | --- |
| Content | Name | Type,length | note |
| ID of train | Id | Integer(11) unsigned | PK |
| go\_time | go\_time | Varchar(50) | Mandatory field |
| Arrival\_time | Arrival\_time | Varchar(50) | Mandatory field |
| routes\_time | routes\_time | Varchar(50) | Mandatory field |
| type\_train\_id | type\_train\_id | Integer(11) unsigned | FK to type\_train |
| train\_company\_id | train\_company\_id | Integer(11) unsigned | FK to train\_companies |

Table Type\_train

|  |  |  |  |
| --- | --- | --- | --- |
| Content | Name | Type,length | note |
| ID of Type\_train | Id | Integer(11) unsigned | PK |
| name | name | Varchar(50) | Mandatory field |
| description | description | Varchar(255) | Mandatory field |

Table Train\_company

|  |  |  |  |
| --- | --- | --- | --- |
| Content | Name | Type,length | note |
| ID of Train\_company | Id | Integer(11) unsigned | PK |
| short\_name | short\_name | Varchar(50) | Mandatory field |
| full\_name | full\_name | Varchar(255) | Mandatory field |

Table Time\_table

|  |  |  |  |
| --- | --- | --- | --- |
| Content | Name | Type,length | note |
| ID of Time\_table | Id | Integer(11) unsigned | PK |
| train\_id | train\_id | Integer(11) unsigned | FK to trains |
| station\_id | station\_id | Integer(11) unsigned | FK to stations |
| go\_time | go\_time | Varchar(50) | Mandatory field |
| arrival\_time | arrival\_time | Varchar(50) | Mandatory field |

Table Routes

|  |  |  |  |
| --- | --- | --- | --- |
| Content | Name | Type,length | note |
| ID of Routes | Id | Integer(11) unsigned | PK |
| start\_station\_id | start\_station\_id | Integer(11) unsigned | FK to stations |
| end\_station\_id | end\_station\_id | Integer(11) unsigned | FK to stations |

Table Stations

|  |  |  |  |
| --- | --- | --- | --- |
| Content | Name | Type,length | note |
| ID of Stations | Id | Integer(11) unsigned | PK |
| Name | name | Varchar(50) | Mandatory field |
| City | City | Varchar(50) | Mandatory field |
| Location | location | Varchar(50) | Mandatory field |

Table user\_roles

|  |  |  |  |
| --- | --- | --- | --- |
| Content | Name | Type,length | note |
| ID of user\_roles | Id | Integer(11) unsigned | PK |
| user\_id | user\_id | Integer(11) unsigned | FK to users |
| role\_id | role\_id | Integer(11) unsigned | FK to roles |

Course work - 4

|  |
| --- |
| -- Database: sdafdf  -- DROP DATABASE sdafdf;  CREATE DATABASE sdafdf  WITH OWNER = postgres  ENCODING = 'UTF8'  TABLESPACE = pg\_default  LC\_COLLATE = 'Russian\_Russia.1251'  LC\_CTYPE = 'Russian\_Russia.1251'  CONNECTION LIMIT = -1;    CREATE TABLE Users (  user\_id int,  first\_name varchar(50) NOT NULL,  last\_name varchar(50) NOT NULL,  middle\_name varchar(50),  INN varchar(60) NOT NULL,  doc\_type varchar(60)NOT NULL,  doc\_number varchar(60)NOT NULL,  doc\_issue varchar(60)NOT NULL,  doc\_expire varchar(60)NOT NULL,  PRIMARY KEY (user\_id)  );  INSERT INTO Users  VALUES(2001,'Aksakalov' , 'Beksultan', 'Assanovich', '123456789', 'PASSPORT','87878787','06.06.2010','12.12.2050'),  (2002,'Aksakalova' , 'Gaukhar', 'Assanovna', '1234567891', 'PASSPORT','87878788','04.08.2008','12.12.2030'),  (2003,'Aksakalov' , 'Sundet', 'Assanovich', '1234567890', 'PASSPORT','87878789','02.07.2009','12.12.2040');  CREATE TABLE Roles (  role\_id int,  description\_r varchar(255) NOT NULL,  PRIMARY KEY (role\_id)  );  INSERT INTO Roles  VALUES(2101,'Train driver'),  (2102,'Passenger'),  (2103,'Conductor');    CREATE TABLE User\_roles (  user\_role\_id int,  user\_id int,  role\_id int,  PRIMARY KEY (user\_role\_id),  FOREIGN KEY (user\_id) REFERENCES Users(user\_id),  FOREIGN KEY (role\_id) REFERENCES Roles(role\_id)  );  INSERT INTO User\_roles  VALUES(1,2001,2101),  (2,2002,2102),  (3,2003,2103);  CREATE TABLE Type\_train (  type\_train\_id int,  tp\_name varchar(50)NOT NULL,  tp\_description varchar(255)NOT NULL,  PRIMARY KEY (type\_train\_id)  );  INSERT INTO Type\_train  VALUES(2611,'Тулпар-Тальго','При производстве всех маятниковых поездов, выпускаемых ТАЛЬГО, особое внимание уделяется безопасности,  скорости, комфорту, экономии энергии и уменьшению загрязнения окружающей среды.'),  (2612,'Актобе-Шымкент','Поезд едет со скоростью не более 180км час'),  (2613,'Шымкент-Алматы','Поезд имеет только один путь');  CREATE TABLE Stations(  station\_id int,  st\_name varchar(50)NOT NULL,  st\_city varchar(50)NOT NULL,  st\_loaction varchar(50)NOT NULL,  PRIMARY KEY (station\_id)  );  INSERT INTO Stations  VALUES(2311,'Moscow','Aktobe','Aksay microdistrict'),  (2312,'Bekzhan','Shymkent','Samal district'),  (2313,'Almaty-1','Almaty','Abilaykhana 7street');  CREATE TABLE Routes (  routes\_id int,  start\_station\_id int,  end\_station\_id int,  PRIMARY KEY(routes\_id),  FOREIGN KEY(start\_station\_id) REFERENCES Stations(station\_id),  FOREIGN KEY(end\_station\_id) REFERENCES Stations(station\_id)  );  INSERT INTO Routes  VALUES(2411,2311,2311),  (2412,2312,2312),  (2413,2313,2313);  CREATE TABLE Train\_company (  train\_company\_id int,  short\_name varchar(50) NOT NULL,  full\_name varchar(255) NOT NULL,  PRIMARY KEY(train\_company\_id)  );  INSERT INTO Train\_company  VALUES(2711,'TOO','"TOO" Astana railway'),  (2712,'AO','"AO" Aktobe'),  (2713,'IP','"IP" Shymkent');  CREATE TABLE Trains (  train\_id int,  go\_time date NOT NULL,  arrival\_time date NOT NULL,  routes\_id int,  type\_train\_id int,  train\_company\_id int,  PRIMARY KEY (train\_id),  FOREIGN KEY (routes\_id)REFERENCES Routes(routes\_id),  FOREIGN KEY (type\_train\_id)REFERENCES Type\_train(type\_train\_id),  FOREIGN KEY (train\_company\_id)REFERENCES Train\_company(train\_company\_id)  );  INSERT INTO Trains  VALUES(2911,'1-Nov-2016','2-Nov-2016',2411,2611,2711),  (2912,'2-June-2016','3-June-2016',2412,2612,2712),  (2913,'3-October-2016','4-October-2016',2413,2613,2713);  CREATE TABLE Classification\_types (  classif\_id int,  classif\_name varchar(255)NOT NULL,  classif\_description varchar(255) NOT NULL,  PRIMARY KEY (classif\_id)  );  INSERT INTO Classification\_types  VALUES (2511,'Compart','Special room in the passenger railway carriage'),  (2512,'Reserved seat','One of the types of passenger railway carriage in Russia and other CIS countries'),  (2513,'luxury','Double compartments which can be accessories');  CREATE TABLE Places (  place\_id int,  pl\_price int NOT NULL,  train\_id int,  classif\_id int,  PRIMARY KEY (place\_id),  FOREIGN KEY (train\_id) REFERENCES Trains(train\_id),  FOREIGN KEY (classif\_id)REFERENCES Classification\_types(classif\_id)  );  INSERT INTO Places  VALUES(3111,2500,2911,2511),  (3112,3500,2912,2512),  (3113,4500,2913,2513);  CREATE TABLE Compare\_places (  compare\_palce\_id int,  user\_id int,  place\_id int,  PRIMARY KEY (compare\_palce\_id),  FOREIGN KEY (user\_id) REFERENCES Users(user\_id),  FOREIGN KEY (place\_id) REFERENCES Places(place\_id)  );  INSERT INTO Compare\_places  VALUES(2211,2001,3111),  (2212,2002,3112),  (2213,2003,3113);  CREATE TABLE Ticket (  ticket\_id int,  print\_time varchar(50),  place\_id int,  PRIMARY KEY (ticket\_id),  FOREIGN KEY (place\_id) REFERENCES Places(place\_id)  ); |

Course work - 5

|  |
| --- |
| INSERT INTO Ticket  VALUES(4111,'23:00 10-Nov-2016',3111),  (4112,'22:20 11-Jun-2015',3112),  (4113,'21:30 12-May-2015',3113);  CREATE TABLE Orders (  order\_id int,  order\_time varchar(255) NOT NULL,  user\_id int,  ticket\_id int,  PRIMARY KEY(order\_id),  FOREIGN KEY (user\_id) REFERENCES Users(user\_id),  FOREIGN KEY (ticket\_id)REFERENCES Ticket(ticket\_id)  );  INSERT INTO Orders  VALUES(5111,'23:00 10-Nov-2016',2001,4111),  (5112,'22:20 11-Jun-2015',2002,4112),  (5113,'21:30 12-May-2015',2003,4113);  CREATE TABLE Time\_table (  time\_id int,  go\_time date NOT NULL,  arrival\_time date NOT NULL,  train\_id int,  station\_id int,  PRIMARY KEY (time\_id),  FOREIGN KEY (train\_id) REFERENCES Trains(train\_id),  FOREIGN KEY (station\_id) REFERENCES Stations(station\_id)  );  INSERT INTO Time\_table  VALUES(3011,'1-May-2015','2-May-2015',2911,2311),  (3012,'2-May-2015','3-May-2015',2912,2312),  (3013,'3-May-2015','4-May-2015',2913,2313);    UPDATE Time\_table SET arrival\_time='05.12.2016' WHERE time\_id=3011;  ALTER TABLE Time\_table  ADD temp varchar(25);      ALTER TABLE Time\_table  DROP temp; |

**Course work. Part 6**

**Queries** **(DB of individual topic)**

**Goal:** to create at least 10 queries for Course Work DB. First of all, implement queries from the Course Work. Part 1 (paragraph 6). Furthermore, list of queries should include the following points:

1. At least 1 query with LIKE operator to find some values.

2. At least 1 query with one of the aggregation functions.

3. At least 1 query with full syntax of the SELECT statement (in other words, using WHERE, GROUP BY, HAVING, ORDER BY in one query).

4. At least 1 query with subquery (using one of the operators =, <, > and etc.)

5. At least 1 query with subquery (using IN or NOT IN).

**1. At least 1 query with LIKE operator to find some values.**

**Description of queries:**

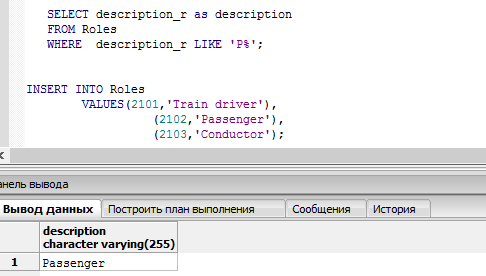
**Scripts:**

SELECT description\_r as description

FROM Roles

WHERE description\_r LIKE 'P%';

**Screenshots** :



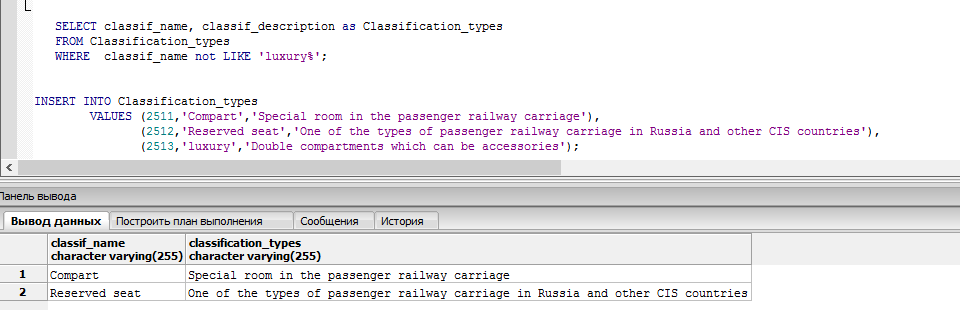
**Description of queries:**

**Scripts:**

**SELECT classif\_name, classif\_description as Classification\_types**

**FROM Classification\_types**

**WHERE classif\_name not LIKE 'luxury%';**

**Screenshots:** 

**2. At least 1 query with one of the aggregation functions.**

**Description of queries:**

**Scripts:**

**SELECT place\_id, pl\_price, train\_id, classif\_id**

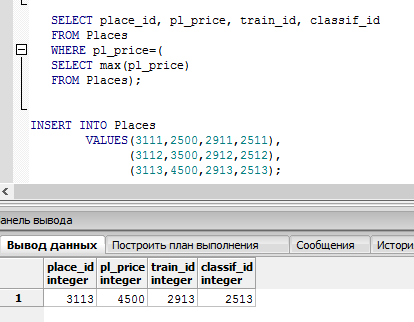
**FROM Places**

**WHERE pl\_price=(**

**SELECT max(pl\_price)**

**FROM Places);**

**Screenshots:**



**Description of queries:**

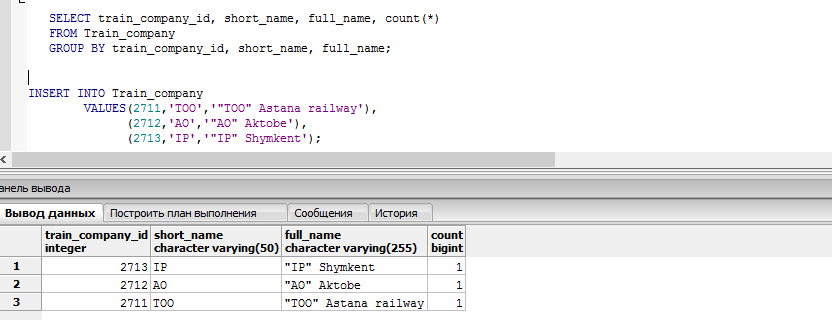
**Scripts:**

**SELECT train\_company\_id, short\_name, full\_name, count(\*)**

**FROM Train\_company**

**GROUP BY train\_company\_id, short\_name, full\_name;**

**Screenshots:**



**3. At least 1 query with full syntax of the SELECT statement (in other words, using WHERE, GROUP BY, HAVING, ORDER BY in one query).**

**Description of queries:**

**Scripts:**

**SELECT us.first\_name||' ' || ord.ticket\_id as description\_passenger, count(\*)**

**FROM Users us, Orders ord**

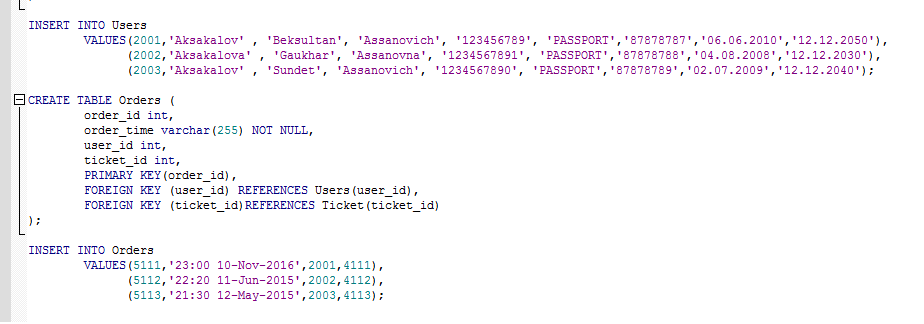
**WHERE us.user\_id = ord.user\_id**

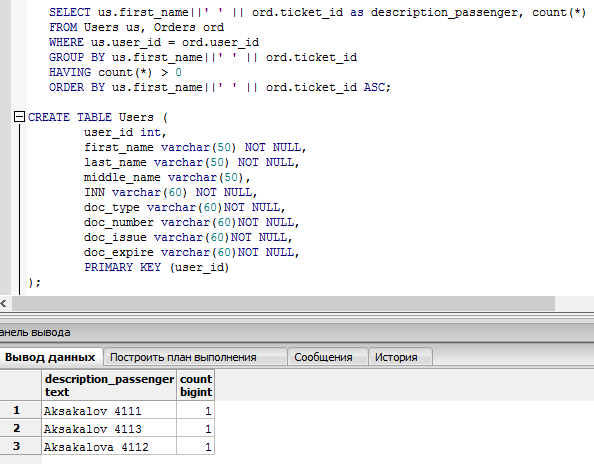
**GROUP BY us.first\_name||' ' || ord.ticket\_id**

**HAVING count(\*) > 0**

**ORDER BY us.first\_name||' ' || ord.ticket\_id ASC;**

**Screenshots:**





**Description of queries:**

**Scripts:**

**SELECT cl.classif\_name||' ' || pl.pl\_price as passenger, count(\*)**

**FROM Classification\_types cl, Places pl**

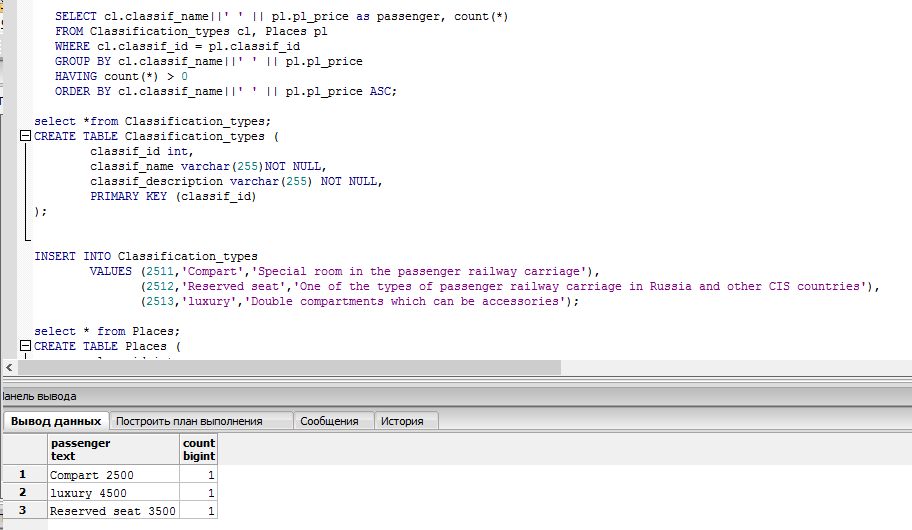
**WHERE cl.classif\_id = pl.classif\_id**

**GROUP BY cl.classif\_name||' ' || pl.pl\_price**

**HAVING count(\*) > 0**

**ORDER BY cl.classif\_name||' ' || pl.pl\_price ASC;**

**Screenshots:**



**4. At least 1 query with subquery (using one of the operators =, <, > and etc.)**

**Description of queries:**

**Scripts:**

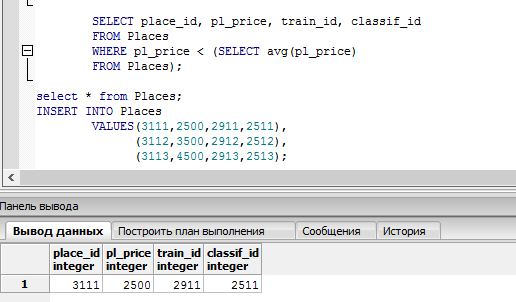
**SELECT place\_id, pl\_price, train\_id, classif\_id**

**FROM Places**

**WHERE pl\_price < (SELECT avg(pl\_price)**

**FROM Places);**

**Screenshots:**



**Description of queries:**

**Scripts:**

**Screenshots:**

**5. At least 1 query with subquery (using IN or NOT IN).**

**SELECT \* FROM users WHERE user\_id in (2001,2002);**



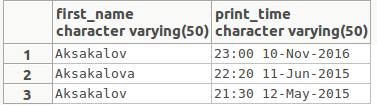
6.

SELECT users.first\_name,ticket.print\_time

FROM users,orders,ticket

WHERE users.user\_id = orders.user\_id AND

ticket.ticket\_id=orders.ticket\_id;

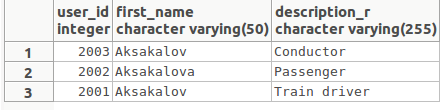


7.SELECT users.user\_id,users.first\_name,roles.description\_r

FROM users,roles,user\_roles

WHERE users.user\_id = user\_roles.user\_id AND

user\_roles.role\_id=roles.role\_id ORDER BY roles.description\_r;



8. SELECT us.middle\_name||' ' || ord.ticket\_id as description\_passenger, count(\*)

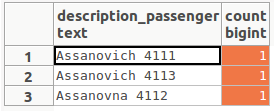
FROM Users us, Orders ord

WHERE us.user\_id = ord.user\_id

GROUP BY us.middle\_name||' ' || ord.ticket\_id

HAVING count(\*) > 0

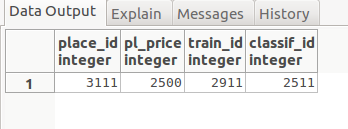
ORDER BY us.middle\_name||' ' || ord.ticket\_id ASC;



9. SELECT \*

FROM Places

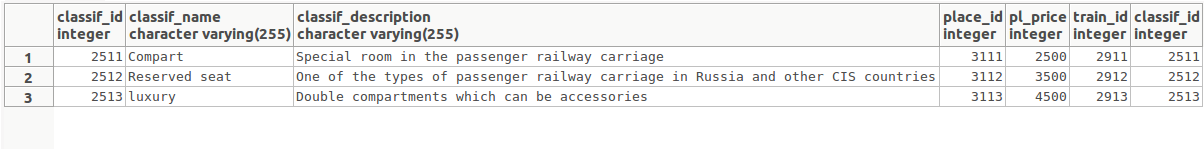
WHERE pl\_price BETWEEN 2500 AND 2880;



10. SELECT \*

FROM Classification\_types cl, Places pl

WHERE cl.classif\_id = pl.classif\_id;



**Conclusion**