UNIVERSITY OF
SOUTH FLORIDA

ISM 6218

Advanced Database Management

Project by Group 8 on

Airline Reservation Database

Group Members:

Deepesh Puthran

Siva Prasad Sahoo

Priyanka Jain

Shruti Pareek

Nikhil P Naik

ACKNOWLEDGEMENT

It is a great pleasure to have the opportunity to extend our heartiest felt gratitude to everybody who helped us throughout the course of this project.

It is distinct pleasure to express our deep sense of gratitude and indebtedness to our learned professor, **Mr. James McCart** for their invaluable guidance and encouragement. With their continuous inspiration only, it becomes possible to complete this Project.

Deepesh Puthran
Siva Prasad Sahoo
Priyanka Jain
Shruti Pareek
Nikhil P Naik

TABLE OF CONTENTS

Contents

1. Introduction

1.1 Overview

Airline reservation system is widely used database system in the world. It is an example of transaction processing systems. Transaction processing systems are systems with large databases and hundreds of concurrent users executing database transactions. These systems should be highly available, and the latency should be low for the hundreds of users accessing it concurrently.

Transaction is the logical unit of database processing, that must be completed in entirety. Typical a transaction includes tasks such as retrieval, insertion, deletion and updates.

In this project we deal with the system that includes the DML parts of the database insertion, deletion and updates along with maintenance and integrity at all the stages of transaction.

The database that we have designed handles the functions of the airline reservation systems such as reservation, cancellation and update of flight trip transaction.

1.2 Feasibility Study & Risk Analysis

It is the most difficult territory to survey since goals, capacities, and execution are hazy; anything appears to be conceivable if right suspicions are made.

Development Risk:

- Can the system that is designed be implemented with the necessary functions and performance.
- Resource availability:
- Are the team members skilled enough to develop the system in hand?

Technology:

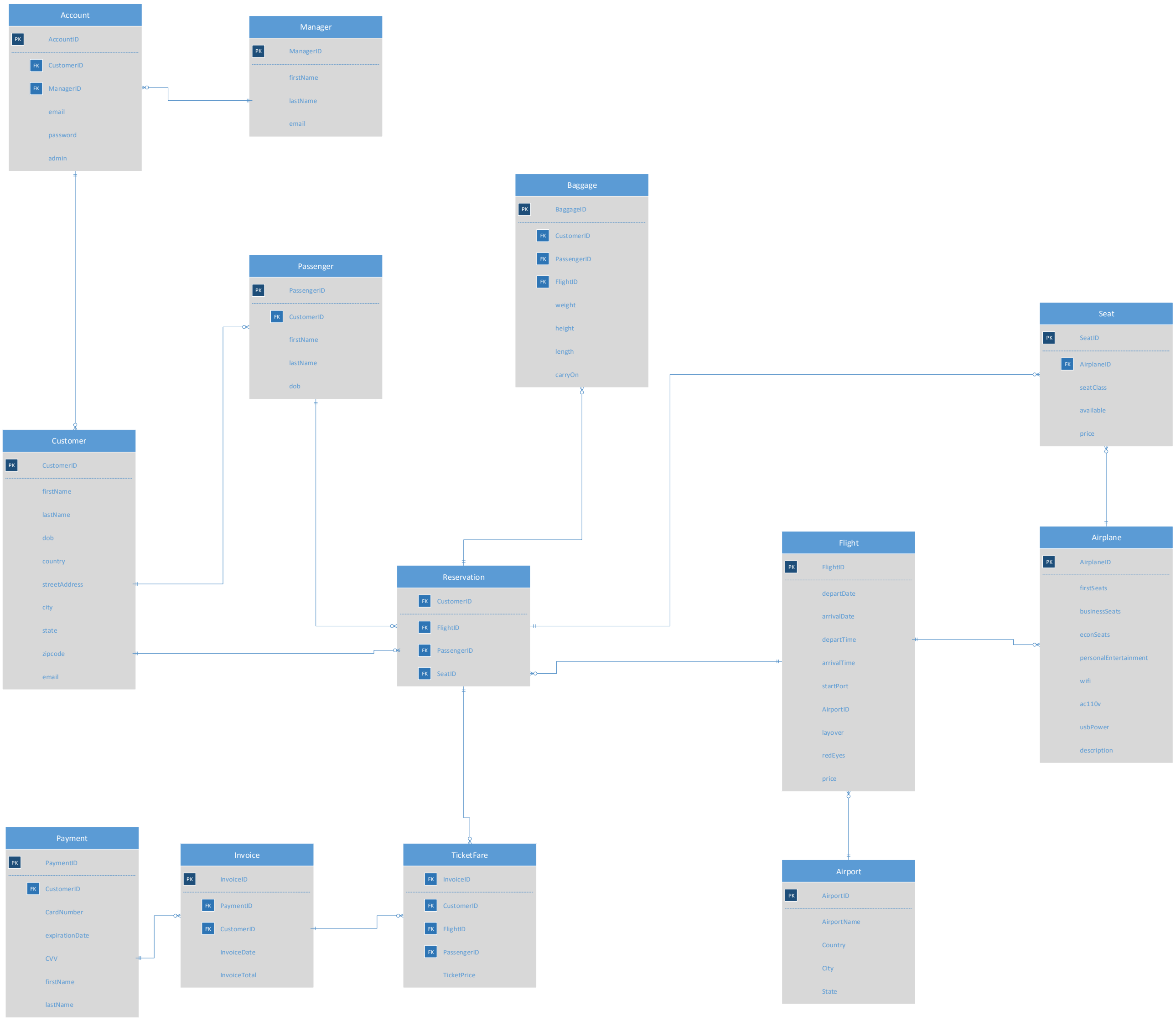
- Has the pertinent innovation advanced to a state that will bolster the framework?
- Most of the above thought additionally applies to the work we have done. To the extent improvements, dangers are concerned, yes essential capacities and the requirements under which they need to perform have been distinguished and separated into modules, so each module plays out its own relegated assignment.

- As far skilled staff is concerned, our team is performing this task we have fully understood the problem. We are proficient enough to finish the task within the given time constraints.
- Here we could also use programming languages such as VB.NET or Java to build a GUI which would help the end user to perform the task more easily but since the report is pertaining to only database design part of the system for which is done using Oracle SQL developer. The tools have got all the functionality.

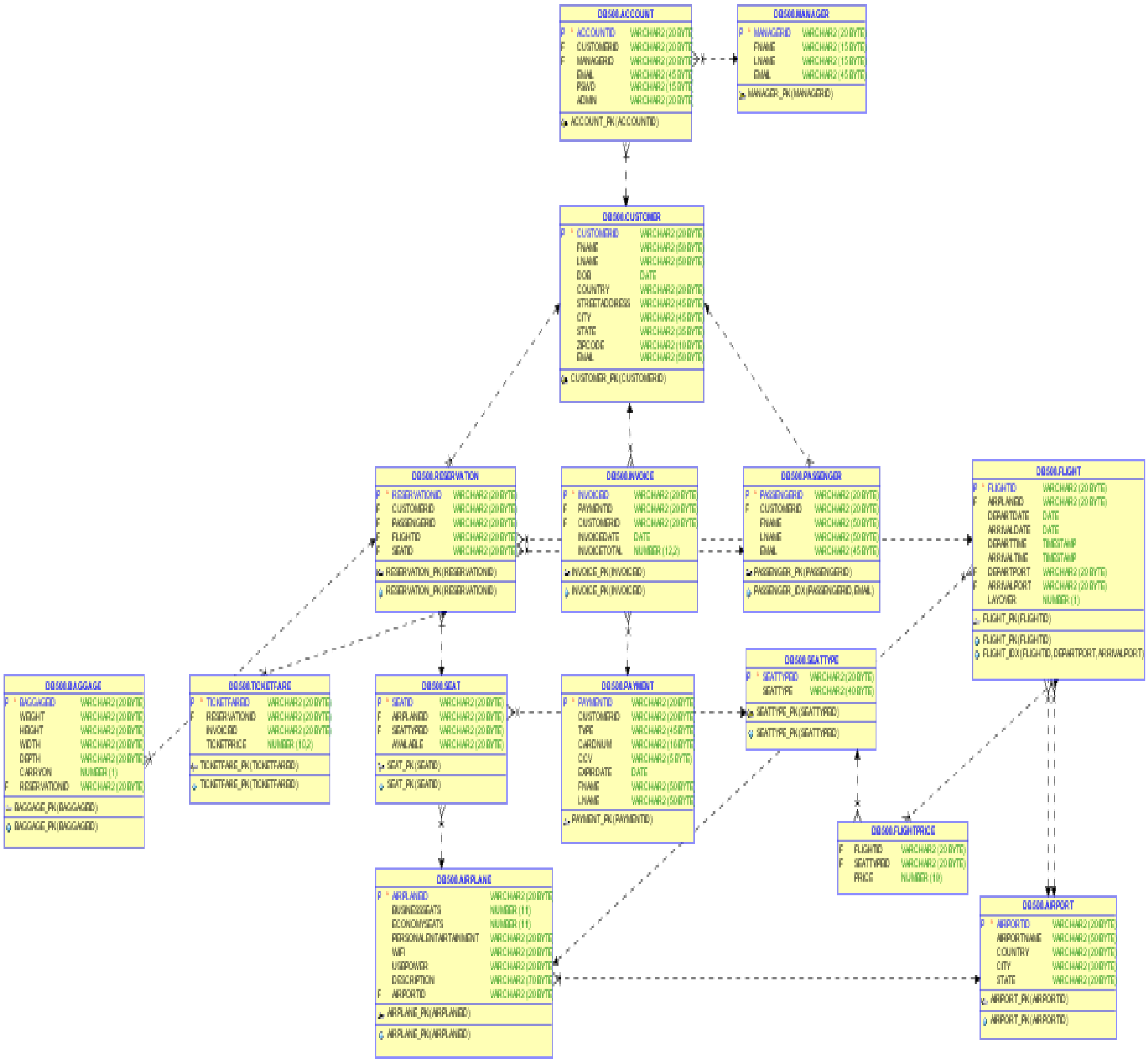
Operational Feasibility:

The study helps us understand whether the system that we are building will be operational with the available staff and with the given time frame. The staff is proficient enough to handle the development of the database system. The design has been done keeping in mind the ease with which user will be able to get the answer of their queries easily. With the use of simple command, menus and proper validations common user will be able to understand the operations of the system.

2. Logical ER-Diagram



3. Physical ER Diagram


DB500.ACCOUNT
P * ACCOUNTID VARCHAR2 (20 BYTE
F CUSTOMERID VARCHAR2 (20 BYTE
F MANAGERID VARCHAR2 (20 BYTE
EMAIL VARCHAR2 (45 BYTE
PSWD VARCHAR2 (15 BYTE
ADMIN VARCHAR2 (20 BYTE
ACCOUNT_PK (ACCOUNTID)
DB500.MANAGER
P * MANAGERID VARCHAR2 (20 BYTE)
FNAME VARCHAR2 (15 BYTE)
LNAME VARCHAR2 (15 BYTE)
EMAIL VARCHAR2 (45 BYTE)
MANAGER_PK (MANAGERID)
DB500.CUSTOMER
P * CUSTOMERID VARCHAR2 (20 BYTE)
FNAME VARCHAR2 (50 BYTE)
LNAME VARCHAR2 (50 BYTE)
DOB DATE
COUNTRY VARCHAR2 (20 BYTE)
STREETADDRESS VARCHAR2 (45 BYTE)
CITY VARCHAR2 (45 BYTE)
STATE VARCHAR2 (35 BYTE)
ZIPCODE VARCHAR2 (10 BYTE)
EMAIL VARCHAR2 (50 BYTE)
CUSTOMER_PK (CUSTOMERID)
DB500.RESERVATION
P * RESERVATIONID VARCHAR2 (20 BYTE)
F CUSTOMERID VARCHAR2 (20 BYTE)
F PASSENGERID VARCHAR2 (20 BYTE)
F FLIGHTID VARCHAR2 (20 BYTE)
F SEATID VARCHAR2 (20 BYTE)
RESERVATION_PK (RESERVATIONID)
RESERVATION_PK (RESERVATIONID)
DB500.INVOICE
P * INVOICEID VARCHAR2 (20 BYTE)
F PAYMENTID VARCHAR2 (20 BYTE)
F CUSTOMERID VARCHAR2 (20 BYTE)
INVOICEDATE DATE
INVOICETOTAL NUMBER (12,2)
INVOICE_PK (INVOICEID)
INVOICE_PK (INVOICEID)
DB500.PASSENGER
P * PASSENGERID VARCHAR2 (20 BYTE)
F CUSTOMERID VARCHAR2 (20 BYTE)
FNAME VARCHAR2 (50 BYTE)
LNAME VARCHAR2 (50 BYTE)
EMAIL VARCHAR2 (45 BYTE)
PASSENGER_PK (PASSENGERID)
PASSENGER_IDX (PASSENGERID, EMAIL)
DB500.FLIGHT
P * FLIGHTID VARCHAR2 (20 BYTE)
F AIRPLANEID VARCHAR2 (20 BYTE)
DEPARTDATE DATE
ARRIVALDATE DATE
DEPARTTIME TIMESTAMP
ARRIVALTIME TIMESTAMP
F DEPARTPORT VARCHAR2 (20 BYTE)
F ARRIVALPORT VARCHAR2 (20 BYTE)
LAYOVER NUMBER (1)
FLIGHT_PK (FLIGHTID)
FLIGHT_PK (FLIGHTID)
FLIGHT_IDX (FLIGHTID, DEPARTPORT, ARRIVALPORT)
DB500.BAGGAGE
P * BAGGAGEID VARCHAR2 (20 BYTE)
WEIGHT VARCHAR2 (20 BYTE)
HEIGHT VARCHAR2 (20 BYTE)
WIDTH VARCHAR2 (20 BYTE)
DEPTH VARCHAR2 (20 BYTE)
CARRYON NUMBER (1)
F RESERVATIONID VARCHAR2 (20 BYTE)
BAGGAGE_PK (BAGGAGEID)
BAGGAGE_PK (BAGGAGEID)
DB500.TICKETFARE
P * TICKETFAREID VARCHAR2 (20 BYTE)
F RESERVATIONID VARCHAR2 (20 BYTE)
INVOICEID VARCHAR2 (20 BYTE)
TICKETPRICE NUMBER (10,2)
TICKETFARE_PK (TICKETFAREID)
TICKETFARE_PK (TICKETFAREID)
DB500.SEAT
P * SEATID VARCHAR2 (20 BYTE)
F AIRPLANEID VARCHAR2 (20 BYTE)
F SEATTYPEID VARCHAR2 (20 BYTE)
AVAILABLE VARCHAR2 (20 BYTE)
SEAT_PK (SEATID)
SEAT_PK (SEATID)
DB500.PAYMENT
P * PAYMENTID VARCHAR2 (20 BYTE
CUSTOMERID VARCHAR2 (20 BYTE
TYPE VARCHAR2 (45 BYTE
CARDNUM VARCHAR2 (16 BYTE
CCV VARCHAR2 (5 BYTE)
EXPIRDATE DATE
FNAME VARCHAR2 (50 BYTE
LNAME VARCHAR2 (50 BYTE
PAYMENT_PK (PAYMENTID)
DB500.SEATTYPE
P * SEATTYPEID VARCHAR2 (20 BYTE)
SEATTYPE VARCHAR2 (40 BYTE)
SEATTYPE_PK (SEATTYPEID)
SEATTYPE_PK (SEATTYPEID)
DB500.FLIGHTPRICE
F FLIGHTID VARCHAR2 (20 BYTE)
F SEATTYPEID VARCHAR2 (20 BYTE)
PRICE NUMBER (10)
DB500.AIRPLANE
P * AIRPLANEID VARCHAR2 (20 BYTE
BUSINESSSEATS NUMBER (11)
ECONOMYSEATS NUMBER (11)
PERSONALENTARTAINMENT VARCHAR2 (20 BYTE
WIFI VARCHAR2 (20 BYTE
USBPOWER VARCHAR2 (20 BYTE
DESCRIPTION VARCHAR2 (70 BYTE
F AIRPORTID VARCHAR2 (20 BYTE
AIRPLANE_PK (AIRPLANEID)
AIRPLANE_PK (AIRPLANEID)
DB500.AIRPORT
P * AIRPORTID VARCHAR2 (20 BYTE)
AIRPORTNAME VARCHAR2 (50 BYTE)
COUNTRY VARCHAR2 (20 BYTE)
CITY VARCHAR2 (30 BYTE)
STATE VARCHAR2 (20 BYTE)
AIRPORT_PK (AIRPORTID)
AIRPORT_PK (AIRPORTID)

4. Tables and Primary Keys

The final table that we have created after normalization are as follows. The primary key of the table is carefully created to make sure that none of the attributes of the table is not dependent on the entity except the primary key. The table are as follows:

- ACCOUNT – <u>AccountID</u>
- AIRPLANE – AirplaneID
- AIRPORT – AirportID
- BAGGAGE – BaggageID
- CUSTOMER – CustomerID
- FLIGHT – FlightID
- FLIGHTPRICE
- INVOICE – InvoiceID
- MANAGER- ManagerID
- PASSENGER – PassengerID
- PAYMENT – PaymentID
- RESERVATION – ReservationID
- SEAT – SeatID
- SEATTYPE – SeattypeID
- TICKETFARE – TicketfareID

5. Queries

Create Table Scripts:

- <u>ACCOUNT</u>

```
CREATE TABLE "DB508"."TICKETFARE" (
    "TICKETFAREID"    VARCHAR2(20 BYTE)
        NOT NULL ENABLE,
    "RESERVATIONID"   VARCHAR2(20 BYTE),
    "INVOICEID"       VARCHAR2(20 BYTE),
    "TICKETPRICE"     NUMBER(10,2),
    CONSTRAINT "TICKETFARE_PK" PRIMARY KEY ( "TICKETFAREID" )
        USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS"
    ENABLE,
    CONSTRAINT "TICKETFARE_FK1" FOREIGN KEY ( "RESERVATIONID" )
        REFERENCES "DB508"."RESERVATION" ( "RESERVATIONID" )
    ENABLE
)
    SEGMENT CREATION IMMEDIATE
        PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS";
```

- <u>AIRPLANE</u>

```
CREATE TABLE "DB508"."AIRPLANE" (
    "AIRPLANEID"               VARCHAR2(20 BYTE)
        NOT NULL ENABLE,
    "BUSINESSSEATS"            NUMBER(11,0),
    "ECONOMYSEATS"             NUMBER(11,0),
    "PERSONALENTAIRTAINMENT"   VARCHAR2(20 BYTE),
    "WIFI"                     VARCHAR2(20 BYTE),
    "USBPOWER"                 VARCHAR2(20 BYTE),
    "DESCRIPTION"              VARCHAR2(70 BYTE),
    "AIRPORTID"                VARCHAR2(20 BYTE),
    CONSTRAINT "AIRPLANE_PK" PRIMARY KEY ( "AIRPLANEID" )
        USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS"
    ENABLE,
    CONSTRAINT "AIRPLANE_FK1" FOREIGN KEY ( "AIRPORTID" )
        REFERENCES "DB508"."AIRPORT" ( "AIRPORTID" )
    ENABLE
)
    SEGMENT CREATION IMMEDIATE
        PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS";
```

- AIRPORT

```
CREATE TABLE "DB508"."AIRPORT" (
    "AIRPORTID"     VARCHAR2(20 BYTE)
        NOT NULL ENABLE,
    "AIRPORTNAME"   VARCHAR2(50 BYTE),
    "COUNTRY"       VARCHAR2(20 BYTE),
    "CITY"          VARCHAR2(30 BYTE),
    "STATE"         VARCHAR2(20 BYTE),
    CONSTRAINT "AIRPORT_PK" PRIMARY KEY ( "AIRPORTID" )
        USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS"
    ENABLE
)
    SEGMENT CREATION IMMEDIATE
        PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS";
```

- BAGGAGE

```
CREATE TABLE "DB508"."BAGGAGE"
   (    "BAGGAGEID" VARCHAR2(20 BYTE) NOT NULL ENABLE,
    "WEIGHT" VARCHAR2(20 BYTE),
    "HEIGHT" VARCHAR2(20 BYTE),
    "WIDTH" VARCHAR2(20 BYTE),
    "DEPTH" VARCHAR2(20 BYTE),
    "CARRYON" NUMBER(1,0),
    "RESERVATIONID" VARCHAR2(20 BYTE),
     CONSTRAINT "BAGGAGE_PK" PRIMARY KEY ("BAGGAGEID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "STUDENTS"  ENABLE,
     CONSTRAINT "BAGGAGE_FK1" FOREIGN KEY ("RESERVATIONID")
      REFERENCES "DB508"."RESERVATION" ("RESERVATIONID") ENABLE
   ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "STUDENTS" ;
```

- CUSTOMER

```
CREATE TABLE "DB508"."CUSTOMER" (
    "CUSTOMERID"       VARCHAR2(20 BYTE),
    "FNAME"            VARCHAR2(50 BYTE),
    "LNAME"            VARCHAR2(50 BYTE),
    "DOB"              DATE,
    "COUNTRY"          VARCHAR2(20 BYTE),
    "STREETADDRESS"    VARCHAR2(45 BYTE),
    "CITY"             VARCHAR2(45 BYTE),
    "STATE"            VARCHAR2(35 BYTE),
    "ZIPCODE"          VARCHAR2(10 BYTE),
    "EMAIL"            VARCHAR2(50 BYTE),
    PRIMARY KEY ( "CUSTOMERID" )
        USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS"
    ENABLE
)
    SEGMENT CREATION IMMEDIATE
        PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS";
```

- FLIGHT

```
CREATE TABLE "DB508"."FLIGHT" (
    "FLIGHTID"       VARCHAR2(20 BYTE)
        NOT NULL ENABLE,
    "AIRPLANEID"     VARCHAR2(20 BYTE),
    "DEPARTDATE"     DATE,
    "ARRIVALDATE"    DATE,
    "DEPARTTIME"     TIMESTAMP(6),
    "ARRIVALTIME"    TIMESTAMP(6),
    "DEPARTPORT"     VARCHAR2(20 BYTE),
    "ARRIVALPORT"    VARCHAR2(20 BYTE),
    "LAYOVER"        NUMBER(1,0),
    CONSTRAINT "FLIGHT_PK" PRIMARY KEY ( "FLIGHTID" )
        USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS"
    ENABLE,
    CONSTRAINT "FLIGHT_FK1" FOREIGN KEY ( "AIRPLANEID" )
        REFERENCES "DB508"."AIRPLANE" ( "AIRPLANEID" )
```

```
            ON DELETE CASCADE
    ENABLE,
    CONSTRAINT "FLIGHT_FK2" FOREIGN KEY ( "DEPARTPORT" )
        REFERENCES "DB508"."AIRPORT" ( "AIRPORTID" )
            ON DELETE CASCADE
    ENABLE,
    CONSTRAINT "FLIGHT_FK3" FOREIGN KEY ( "ARRIVALPORT" )
        REFERENCES "DB508"."AIRPORT" ( "AIRPORTID" )
            ON DELETE CASCADE
    ENABLE
)
    SEGMENT CREATION IMMEDIATE
        PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS";

CREATE INDEX "DB508"."FLIGHT_IDX" ON
    "DB508"."FLIGHT" (
        "FLIGHTID",
        "DEPARTPORT",
        "ARRIVALPORT"
    )
        PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS";
```

- <u>FLIGHTPRICE</u>

```
CREATE TABLE "DB508"."FLIGHTPRICE" (
    "FLIGHTID"      VARCHAR2(20 BYTE),
    "SEATTYPEID"    VARCHAR2(20 BYTE),
    "PRICE"         NUMBER(10,0),
    CONSTRAINT "FLIGHTPRICE_FK1" FOREIGN KEY ( "SEATTYPEID" )
        REFERENCES "DB508"."SEATTYPE" ( "SEATTYPEID" )
            ON DELETE CASCADE
    ENABLE,
    CONSTRAINT "FLIGHTPRICE_FK2" FOREIGN KEY ( "FLIGHTID" )
        REFERENCES "DB508"."FLIGHT" ( "FLIGHTID" )
    ENABLE
)
    SEGMENT CREATION IMMEDIATE
        PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS";
```

- INVOICE

```
CREATE TABLE "DB508"."INVOICE" (
    "INVOICEID"      VARCHAR2(20 BYTE)
        NOT NULL ENABLE,
    "PAYMENTID"      VARCHAR2(20 BYTE),
    "CUSTOMERID"     VARCHAR2(20 BYTE),
    "INVOICEDATE"    DATE,
    "INVOICETOTAL"   NUMBER(12,2),
    CONSTRAINT "INVOICE_PK" PRIMARY KEY ( "INVOICEID" )
        USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS"
    ENABLE,
    CONSTRAINT "INVOICE_FK1" FOREIGN KEY ( "CUSTOMERID" )
        REFERENCES "DB508"."CUSTOMER" ( "CUSTOMERID" )
            ON DELETE CASCADE
    ENABLE,
    CONSTRAINT "INVOICE_FK2" FOREIGN KEY ( "PAYMENTID" )
        REFERENCES "DB508"."PAYMENT" ( "PAYMENTID" )
            ON DELETE CASCADE
    ENABLE
)
    SEGMENT CREATION IMMEDIATE
        PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS";
```

- MANAGER

```
CREATE TABLE "DB508"."MANAGER" (
    "MANAGERID"   VARCHAR2(20 BYTE),
    "FNAME"       VARCHAR2(15 BYTE),
    "LNAME"       VARCHAR2(15 BYTE),
    "EMAIL"       VARCHAR2(45 BYTE),
    PRIMARY KEY ( "MANAGERID" )
        USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS"
    ENABLE
)
    SEGMENT CREATION IMMEDIATE
        PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS";
```

- PASSENGER

```
CREATE TABLE "DB508"."PASSENGER" (
    "PASSENGERID"   VARCHAR2(20 BYTE),
    "CUSTOMERID"    VARCHAR2(20 BYTE),
    "FNAME"         VARCHAR2(50 BYTE),
    "LNAME"         VARCHAR2(50 BYTE),
    "EMAIL"         VARCHAR2(45 BYTE),
    PRIMARY KEY ( "PASSENGERID" )
        USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS"
    ENABLE,
    CONSTRAINT "PASSENGER_FK1" FOREIGN KEY ( "CUSTOMERID" )
        REFERENCES "DB508"."CUSTOMER" ( "CUSTOMERID" )
    ENABLE
)
    SEGMENT CREATION IMMEDIATE
        PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS";

CREATE INDEX "DB508"."PASSENGER_IDX" ON
    "DB508"."PASSENGER" (
        "PASSENGERID",
        "EMAIL"
    )
        PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS";
```

- PAYMENT

```
 CREATE TABLE "DB508"."PAYMENT" (

    "PAYMENTID"     VARCHAR2(20 BYTE),
    "CUSTOMERID"    VARCHAR2(20 BYTE),
    "TYPE"          VARCHAR2(45 BYTE),
    "CARDNUM"       VARCHAR2(16 BYTE),
    "CCV"           VARCHAR2(5 BYTE),
    "EXPIRDATE"     DATE,
    "FNAME"         VARCHAR2(50 BYTE),
    "LNAME"         VARCHAR2(50 BYTE),
    PRIMARY KEY ( "PAYMENTID" )
        USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
```

```
        TABLESPACE "STUDENTS"
    ENABLE
)
    SEGMENT CREATION IMMEDIATE
        PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS";
```

- RESERVATION

```
 CREATE TABLE "DB508"."RESERVATION" (

    "RESERVATIONID"   VARCHAR2(20 BYTE)
        NOT NULL ENABLE,
    "CUSTOMERID"      VARCHAR2(20 BYTE),
    "PASSENGERID"     VARCHAR2(20 BYTE),
    "FLIGHTID"        VARCHAR2(20 BYTE),
    "SEATID"          VARCHAR2(20 BYTE),
    CONSTRAINT "RESERVATION_PK" PRIMARY KEY ( "RESERVATIONID" )
        USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS"
    ENABLE,
    CONSTRAINT "RESERVATION_FK2" FOREIGN KEY ( "PASSENGERID" )
        REFERENCES "DB508"."PASSENGER" ( "PASSENGERID" )
            ON DELETE CASCADE
    ENABLE,
    CONSTRAINT "RESERVATION_FK3" FOREIGN KEY ( "CUSTOMERID" )
        REFERENCES "DB508"."CUSTOMER" ( "CUSTOMERID" )
    ENABLE,
    CONSTRAINT "RESERVATION_FK4" FOREIGN KEY ( "SEATID" )
        REFERENCES "DB508"."SEAT" ( "SEATID" )
    ENABLE,
    CONSTRAINT "RESERVATION_FK1" FOREIGN KEY ( "FLIGHTID" )
        REFERENCES "DB508"."FLIGHT" ( "FLIGHTID" )
            ON DELETE CASCADE
    ENABLE
)
    SEGMENT CREATION IMMEDIATE
        PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS";
```

- <u>SEAT</u>

```
 CREATE TABLE "DB508"."SEAT" (

    "SEATID"       VARCHAR2(20 BYTE)
        NOT NULL ENABLE,
    "AIRPLANEID"   VARCHAR2(20 BYTE),
    "SEATTYPEID"   VARCHAR2(20 BYTE),
    "AVAILABLE"    VARCHAR2(20 BYTE),
    CONSTRAINT "SEAT_PK" PRIMARY KEY ( "SEATID" )
        USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS"
    ENABLE,
    CONSTRAINT "SEAT_FK1" FOREIGN KEY ( "AIRPLANEID" )
        REFERENCES "DB508"."AIRPLANE" ( "AIRPLANEID" )
            ON DELETE CASCADE
    ENABLE,
    CONSTRAINT "SEAT_FK2" FOREIGN KEY ( "SEATTYPEID" )
        REFERENCES "DB508"."SEATTYPE" ( "SEATTYPEID" )
            ON DELETE CASCADE
    ENABLE
)
    SEGMENT CREATION IMMEDIATE
        PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS";
```

- <u>SEATTYPE</u>

```
CREATE TABLE "DB508"."SEATTYPE" (
    "SEATTYPEID"   VARCHAR2(20 BYTE) NULL ENABLE,
    "SEATTYPE"     VARCHAR2(40 BYTE),
    CONSTRAINT "SEATTYPE_PK" PRIMARY KEY ( "SEATTYPEID" )
        USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS"
    ENABLE
)
    SEGMENT CREATION IMMEDIATE
        PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
            STORAGE ( INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT )
        TABLESPACE "STUDENTS";
```

- TICKETFARE

```
CREATE TABLE "DB508"."TICKETFARE"
   (
        "TICKETFAREID" VARCHAR2(20 BYTE) NOT NULL ENABLE,
        "RESERVATIONID" VARCHAR2(20 BYTE),
        "INVOICEID" VARCHAR2(20 BYTE),
        "TICKETPRICE" NUMBER(10,2),
        CONSTRAINT "TICKETFARE_PK" PRIMARY KEY ("TICKETFAREID")
        USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
        STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645
        PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
        BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
        TABLESPACE "STUDENTS"  ENABLE,
        CONSTRAINT "TICKETFARE_FK1" FOREIGN KEY ("RESERVATIONID")
        REFERENCES "DB508"."RESERVATION" ("RESERVATIONID") ENABLE
   )SEGMENT CREATION IMMEDIATE
    PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
    NOCOMPRESS LOGGING
    STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
    PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
    BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
    TABLESPACE "STUDENTS" ;
```

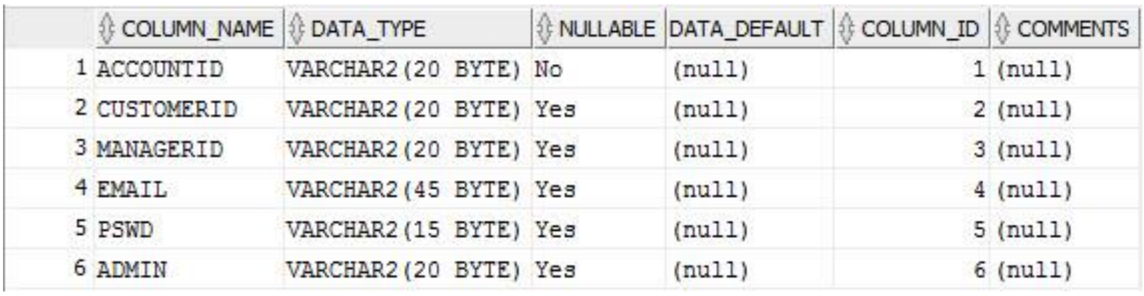
6. Data Generation and Loading:

We have used website www.mockaroo.com for data generation. We downloaded excel files containing the data and then imported them in the tables that we created. We have a total of 32829 records in the database.

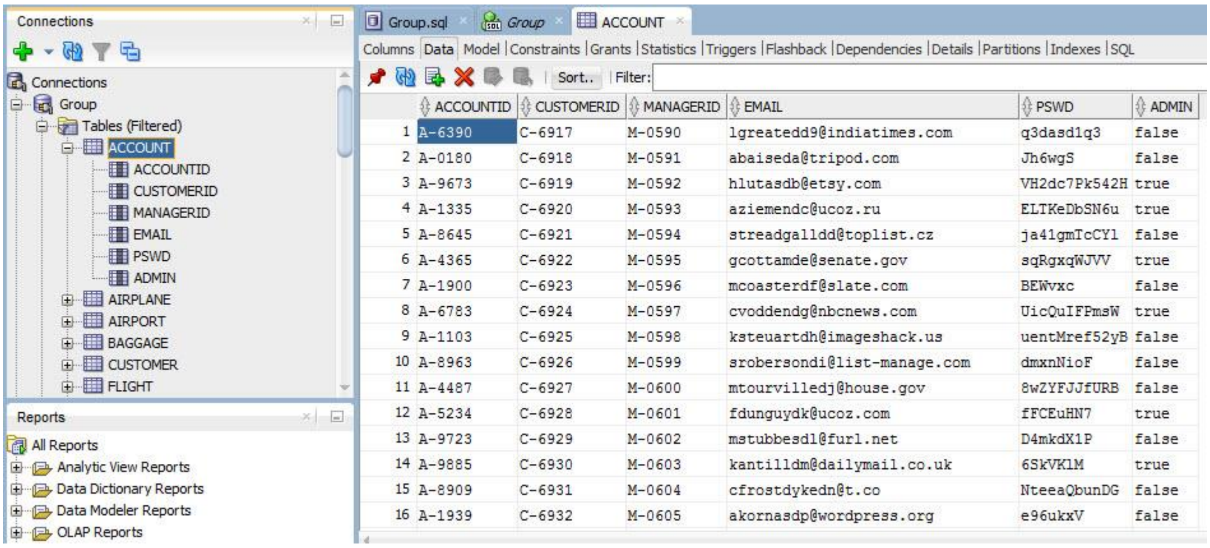
Below is the number of records per table:

1. ACCOUNT- 900
2. AIRPLANE- 1979
3. AIRPORT- 926
4. BAGGAGE- 2000
5. CUSTOMER- 10000
6. FLIGHT- 610
7. FLIGHTPRICE- 2000
8. INVOICE- 3319
9. MANAGER- 1000
10. PASSENGER- 3000
11. PAYMENT- 4000
12. RESERVATION-1000
13. SEAT- 93
14. SEATTYPE- 2
15. TICKETFARE- 2000

1. ACCOUNT

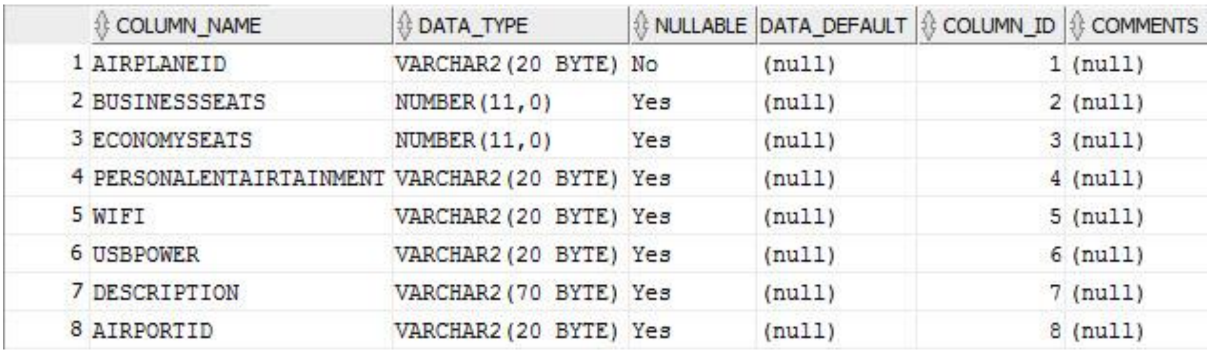


	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ACCOUNTID	VARCHAR2(20 BYTE)	No	(null)	1	(null)
2	CUSTOMERID	VARCHAR2(20 BYTE)	Yes	(null)	2	(null)
3	MANAGERID	VARCHAR2(20 BYTE)	Yes	(null)	3	(null)
4	EMAIL	VARCHAR2(45 BYTE)	Yes	(null)	4	(null)
5	PSWD	VARCHAR2(15 BYTE)	Yes	(null)	5	(null)
6	ADMIN	VARCHAR2(20 BYTE)	Yes	(null)	6	(null)

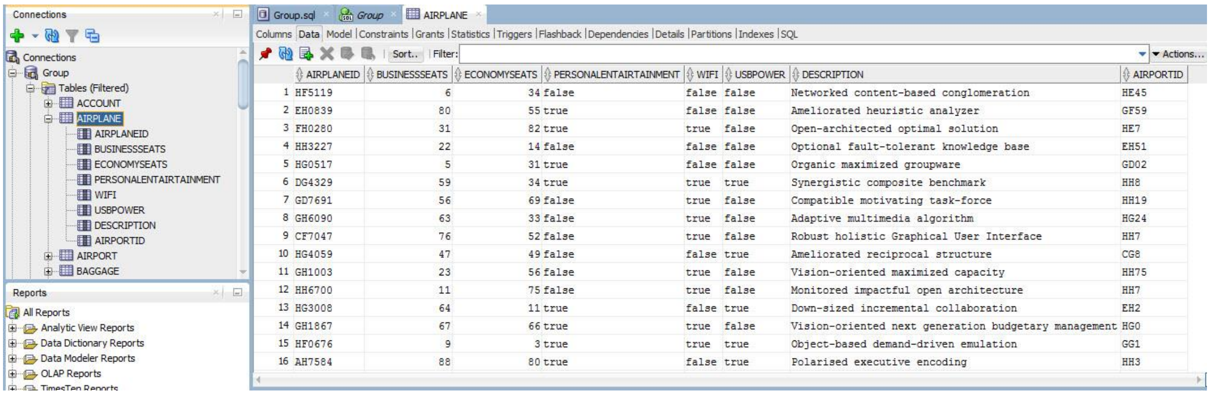


	ACCOUNTID	CUSTOMERID	MANAGERID	EMAIL	PSWD	ADMIN
1	A-6390	C-6917	M-0590	lgreatedd9@indiatimes.com	q3dasd1q3	false
2	A-0180	C-6918	M-0591	abaiseda@tripod.com	Jh6wgS	false
3	A-9673	C-6919	M-0592	hlutasdb@etsy.com	VH2dc7Pk542H	true
4	A-1335	C-6920	M-0593	aziemendc@ucoz.ru	ELTKeDbSN6u	true
5	A-8645	C-6921	M-0594	streadgalldd@toplist.cz	ja41gmTcCY1	false
6	A-4365	C-6922	M-0595	gcottamde@senate.gov	sqRgxqWJVV	true
7	A-1900	C-6923	M-0596	mcoasterdf@slate.com	BEWvxc	false
8	A-6783	C-6924	M-0597	cvoddendg@nbcnews.com	UicQuIFPmsW	true
9	A-1103	C-6925	M-0598	ksteuartdh@imageshack.us	uentMref52yB	false
10	A-8963	C-6926	M-0599	srobersondi@list-manage.com	dmxnNioF	false
11	A-4487	C-6927	M-0600	mtourvilledj@house.gov	8wZYFJJfURB	false
12	A-5234	C-6928	M-0601	fdunguydk@ucoz.com	fFCEuHN7	true
13	A-9723	C-6929	M-0602	mstubbesdl@furl.net	D4mkdX1P	false
14	A-9885	C-6930	M-0603	kantilldm@dailymail.co.uk	6SkVKlM	true
15	A-8909	C-6931	M-0604	cfrostdykedn@t.co	NteeaQbunDG	false
16	A-1939	C-6932	M-0605	akornasdp@wordpress.org	e96ukxV	false

2. AIRPLANE

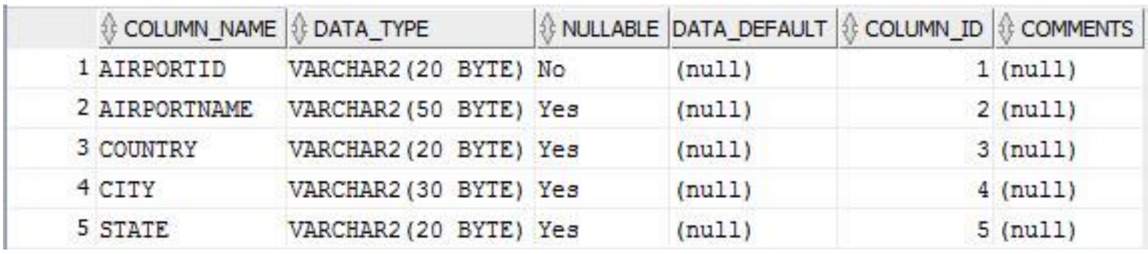


	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	AIRPLANEID	VARCHAR2(20 BYTE)	No	(null)	1	(null)
2	BUSINESSSEATS	NUMBER(11,0)	Yes	(null)	2	(null)
3	ECONOMYSEATS	NUMBER(11,0)	Yes	(null)	3	(null)
4	PERSONALENTAIRTAINMENT	VARCHAR2(20 BYTE)	Yes	(null)	4	(null)
5	WIFI	VARCHAR2(20 BYTE)	Yes	(null)	5	(null)
6	USBPOWER	VARCHAR2(20 BYTE)	Yes	(null)	6	(null)
7	DESCRIPTION	VARCHAR2(70 BYTE)	Yes	(null)	7	(null)
8	AIRPORTID	VARCHAR2(20 BYTE)	Yes	(null)	8	(null)

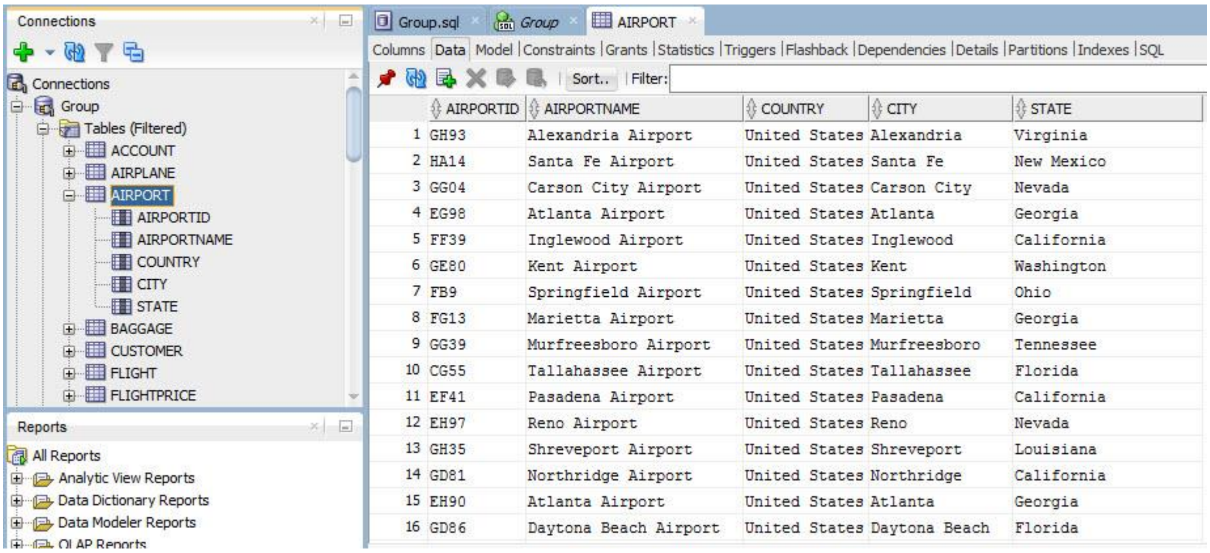


	AIRPLANEID	BUSINESSSEATS	ECONOMYSEATS	PERSONALENTAIRTAINMENT	WIFI	USBPOWER	DESCRIPTION	AIRPORTID
1	HF5119	6	34	false	false	false	Networked content-based conglomeration	HE45
2	EH0839	80	55	true	false	false	Ameliorated heuristic analyzer	GF59
3	FH0280	31	82	true	true	false	Open-architected optimal solution	HE7
4	HH3227	22	14	false	false	false	Optional fault-tolerant knowledge base	EH51
5	HG0517	5	31	true	false	false	Organic maximized groupware	GD02
6	DG4329	59	34	true	true	true	Synergistic composite benchmark	HH8
7	GD7691	56	69	false	true	false	Compatible motivating task-force	HH19
8	GH6090	63	33	false	true	false	Adaptive multimedia algorithm	HG24
9	CF7047	76	52	false	true	false	Robust holistic Graphical User Interface	HH7
10	HG4059	47	49	false	false	true	Ameliorated reciprocal structure	CG8
11	GH1003	23	56	false	true	false	Vision-oriented maximized capacity	HH75
12	HH6700	11	75	false	true	false	Monitored impactful open architecture	HH7
13	HG3008	64	11	true	false	true	Down-sized incremental collaboration	EH2
14	GH1867	67	66	true	true	false	Vision-oriented next generation budgetary management	HG0
15	HF0676	9	3	true	true	true	Object-based demand-driven emulation	GG1
16	AH7584	88	80	true	false	true	Polarised executive encoding	HH3

3. AIRPORT

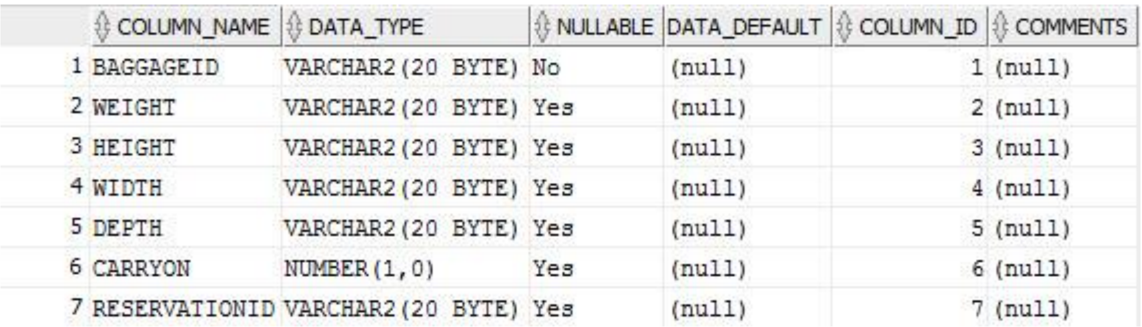


	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	AIRPORTID	VARCHAR2(20 BYTE)	No	(null)	1	(null)
2	AIRPORTNAME	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)
3	COUNTRY	VARCHAR2(20 BYTE)	Yes	(null)	3	(null)
4	CITY	VARCHAR2(30 BYTE)	Yes	(null)	4	(null)
5	STATE	VARCHAR2(20 BYTE)	Yes	(null)	5	(null)

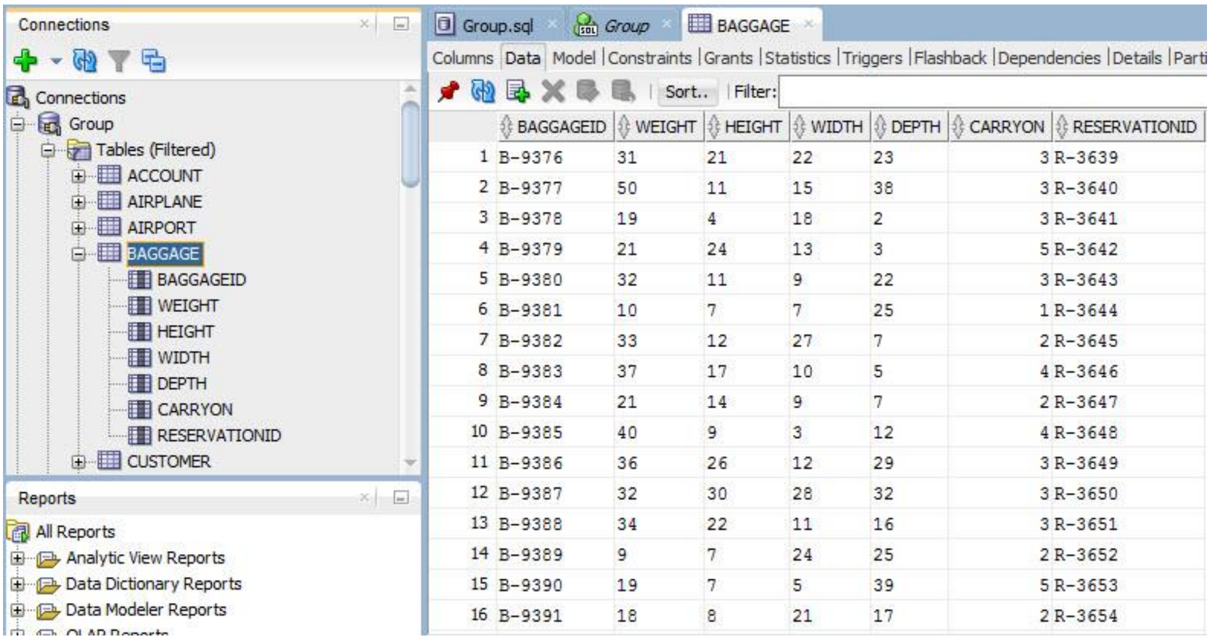


	AIRPORTID	AIRPORTNAME	COUNTRY	CITY	STATE
1	GH93	Alexandria Airport	United States	Alexandria	Virginia
2	HA14	Santa Fe Airport	United States	Santa Fe	New Mexico
3	GG04	Carson City Airport	United States	Carson City	Nevada
4	EG98	Atlanta Airport	United States	Atlanta	Georgia
5	FF39	Inglewood Airport	United States	Inglewood	California
6	GE80	Kent Airport	United States	Kent	Washington
7	FB9	Springfield Airport	United States	Springfield	Ohio
8	FG13	Marietta Airport	United States	Marietta	Georgia
9	GG39	Murfreesboro Airport	United States	Murfreesboro	Tennessee
10	CG55	Tallahassee Airport	United States	Tallahassee	Florida
11	EF41	Pasadena Airport	United States	Pasadena	California
12	EH97	Reno Airport	United States	Reno	Nevada
13	GH35	Shreveport Airport	United States	Shreveport	Louisiana
14	GD81	Northridge Airport	United States	Northridge	California
15	EH90	Atlanta Airport	United States	Atlanta	Georgia
16	GD86	Daytona Beach Airport	United States	Daytona Beach	Florida

4. BAGGAGE

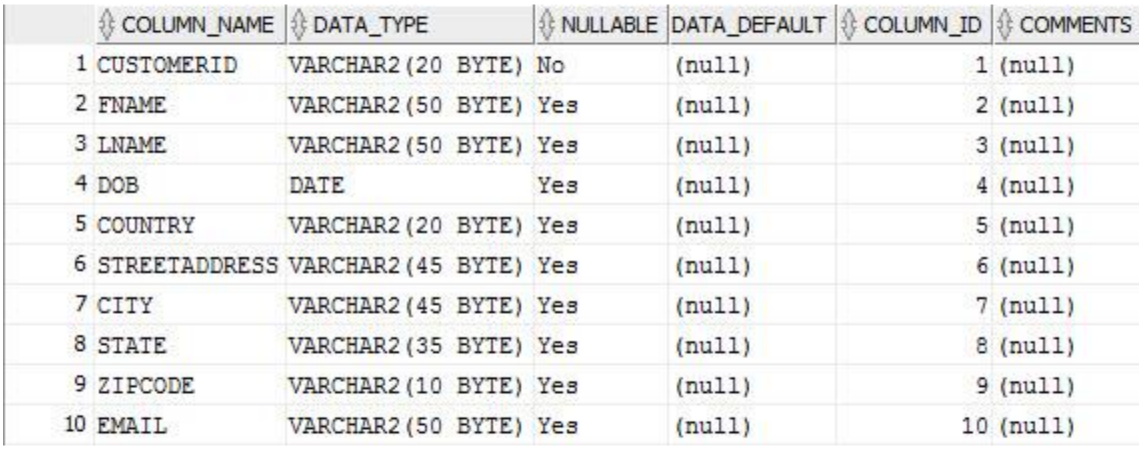


	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	BAGGAGEID	VARCHAR2(20 BYTE)	No	(null)	1	(null)
2	WEIGHT	VARCHAR2(20 BYTE)	Yes	(null)	2	(null)
3	HEIGHT	VARCHAR2(20 BYTE)	Yes	(null)	3	(null)
4	WIDTH	VARCHAR2(20 BYTE)	Yes	(null)	4	(null)
5	DEPTH	VARCHAR2(20 BYTE)	Yes	(null)	5	(null)
6	CARRYON	NUMBER(1,0)	Yes	(null)	6	(null)
7	RESERVATIONID	VARCHAR2(20 BYTE)	Yes	(null)	7	(null)

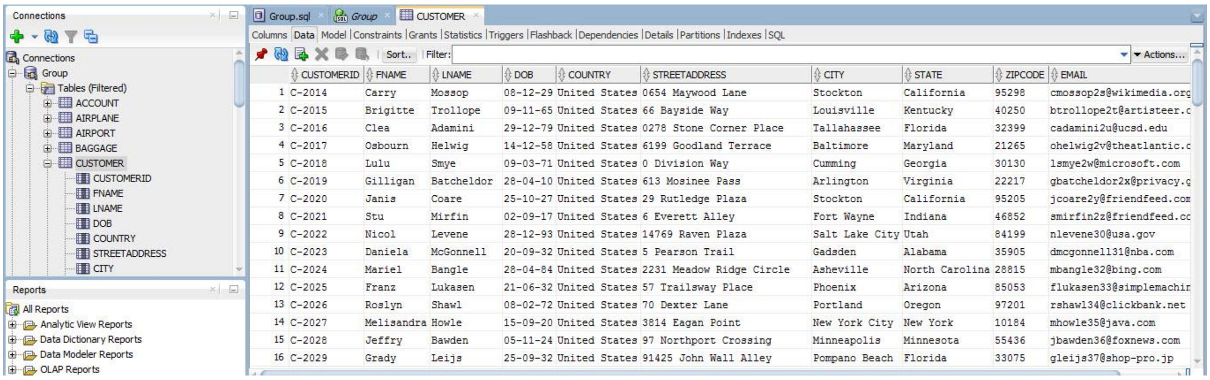


	BAGGAGEID	WEIGHT	HEIGHT	WIDTH	DEPTH	CARRYON	RESERVATIONID
1	B-9376	31	21	22	23	3	R-3639
2	B-9377	50	11	15	38	3	R-3640
3	B-9378	19	4	18	2	3	R-3641
4	B-9379	21	24	13	3	5	R-3642
5	B-9380	32	11	9	22	3	R-3643
6	B-9381	10	7	7	25	1	R-3644
7	B-9382	33	12	27	7	2	R-3645
8	B-9383	37	17	10	5	4	R-3646
9	B-9384	21	14	9	7	2	R-3647
10	B-9385	40	9	3	12	4	R-3648
11	B-9386	36	26	12	29	3	R-3649
12	B-9387	32	30	28	32	3	R-3650
13	B-9388	34	22	11	16	3	R-3651
14	B-9389	9	7	24	25	2	R-3652
15	B-9390	19	7	5	39	5	R-3653
16	B-9391	18	8	21	17	2	R-3654

5. CUSTOMER

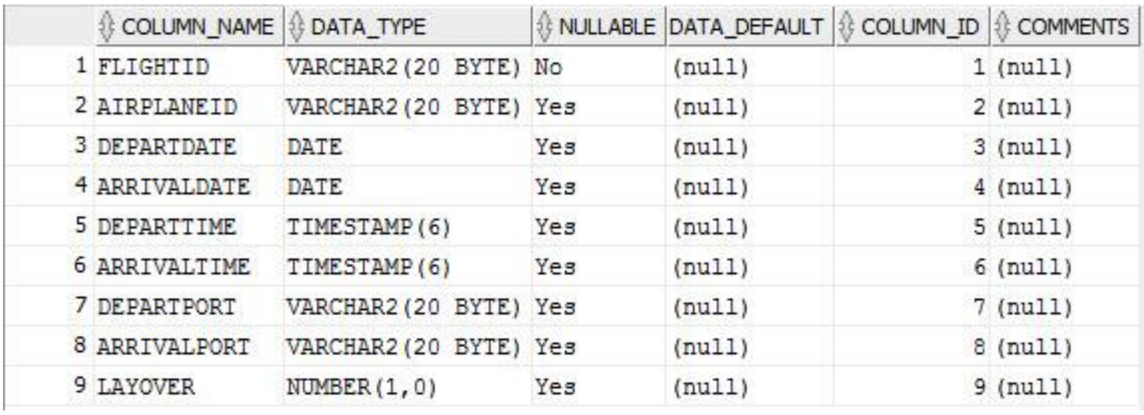


	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	CUSTOMERID	VARCHAR2(20 BYTE)	No	(null)	1	(null)
2	FNAME	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)
3	LNAME	VARCHAR2(50 BYTE)	Yes	(null)	3	(null)
4	DOB	DATE	Yes	(null)	4	(null)
5	COUNTRY	VARCHAR2(20 BYTE)	Yes	(null)	5	(null)
6	STREETADDRESS	VARCHAR2(45 BYTE)	Yes	(null)	6	(null)
7	CITY	VARCHAR2(45 BYTE)	Yes	(null)	7	(null)
8	STATE	VARCHAR2(35 BYTE)	Yes	(null)	8	(null)
9	ZIPCODE	VARCHAR2(10 BYTE)	Yes	(null)	9	(null)
10	EMAIL	VARCHAR2(50 BYTE)	Yes	(null)	10	(null)

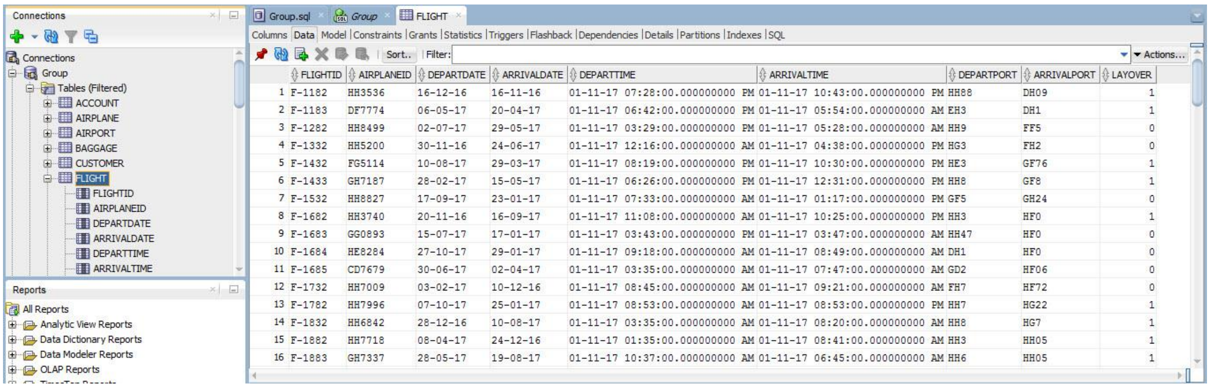


	CUSTOMERID	FNAME	LNAME	DOB	COUNTRY	STREETADDRESS	CITY	STATE	ZIPCODE	EMAIL
1	C-2014	Carry	Mossop	08-12-29	United States	0654 Maywood Lane	Stockton	California	95298	cmossop2s@wikimedia.org
2	C-2015	Brigitte	Trollope	09-11-65	United States	66 Bayside Way	Louisville	Kentucky	40250	btrollope2t@artisteer.c
3	C-2016	Clea	Adamini	29-12-79	United States	0278 Stone Corner Place	Tallahassee	Florida	32399	cadamini2u@ucsd.edu
4	C-2017	Osbourn	Helwig	14-12-58	United States	6199 Goodland Terrace	Baltimore	Maryland	21265	ohelwig2v@theatlantic.c
5	C-2018	Lulu	Smye	09-03-71	United States	0 Division Way	Cumming	Georgia	30130	lsmye2w@microsoft.com
6	C-2019	Gilligan	Batcheldor	28-04-10	United States	613 Mosinee Pass	Arlington	Virginia	22217	gbatcheldor2x@privacy.g
7	C-2020	Janis	Coare	25-10-27	United States	29 Rutledge Plaza	Stockton	California	95205	jcoare2y@friendfeed.com
8	C-2021	Stu	Mirfin	02-09-17	United States	6 Everett Alley	Fort Wayne	Indiana	46852	smirfin2z@friendfeed.cc
9	C-2022	Nicol	Levene	28-12-93	United States	14769 Raven Plaza	Salt Lake City	Utah	84199	nlevene30@usa.gov
10	C-2023	Daniela	McGonnell	20-09-32	United States	5 Pearson Trail	Gadsden	Alabama	35905	dmcgonnell31@nba.com
11	C-2024	Mariel	Bangle	28-04-84	United States	2231 Meadow Ridge Circle	Asheville	North Carolina	28815	mbangle32@bing.com
12	C-2025	Franz	Lukasen	21-06-32	United States	57 Trailsway Place	Phoenix	Arizona	85053	flukasen33@simplemachir
13	C-2026	Roslyn	Shawl	08-02-72	United States	70 Dexter Lane	Portland	Oregon	97201	rshawl34@clickbank.net
14	C-2027	Melisandra	Howle	15-09-20	United States	3814 Eagan Point	New York City	New York	10184	mhowle35@java.com
15	C-2028	Jeffry	Bawden	05-11-24	United States	97 Northport Crossing	Minneapolis	Minnesota	55436	jbawden36@foxnews.com
16	C-2029	Grady	Leijs	25-09-32	United States	91425 John Wall Alley	Pompano Beach	Florida	33075	gleijs37@shop-pro.jp

6. FLIGHT

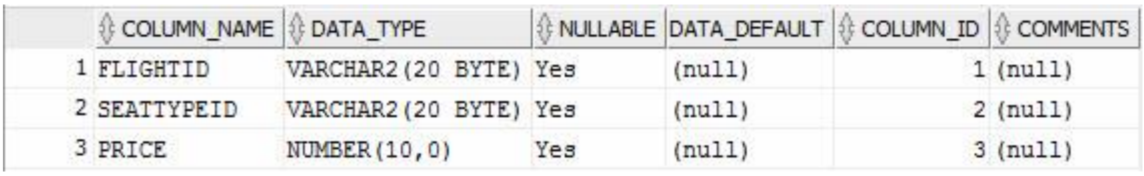


	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	FLIGHTID	VARCHAR2(20 BYTE)	No	(null)	1	(null)
2	AIRPLANEID	VARCHAR2(20 BYTE)	Yes	(null)	2	(null)
3	DEPARTDATE	DATE	Yes	(null)	3	(null)
4	ARRIVALDATE	DATE	Yes	(null)	4	(null)
5	DEPARTTIME	TIMESTAMP(6)	Yes	(null)	5	(null)
6	ARRIVALTIME	TIMESTAMP(6)	Yes	(null)	6	(null)
7	DEPARTPORT	VARCHAR2(20 BYTE)	Yes	(null)	7	(null)
8	ARRIVALPORT	VARCHAR2(20 BYTE)	Yes	(null)	8	(null)
9	LAYOVER	NUMBER(1,0)	Yes	(null)	9	(null)

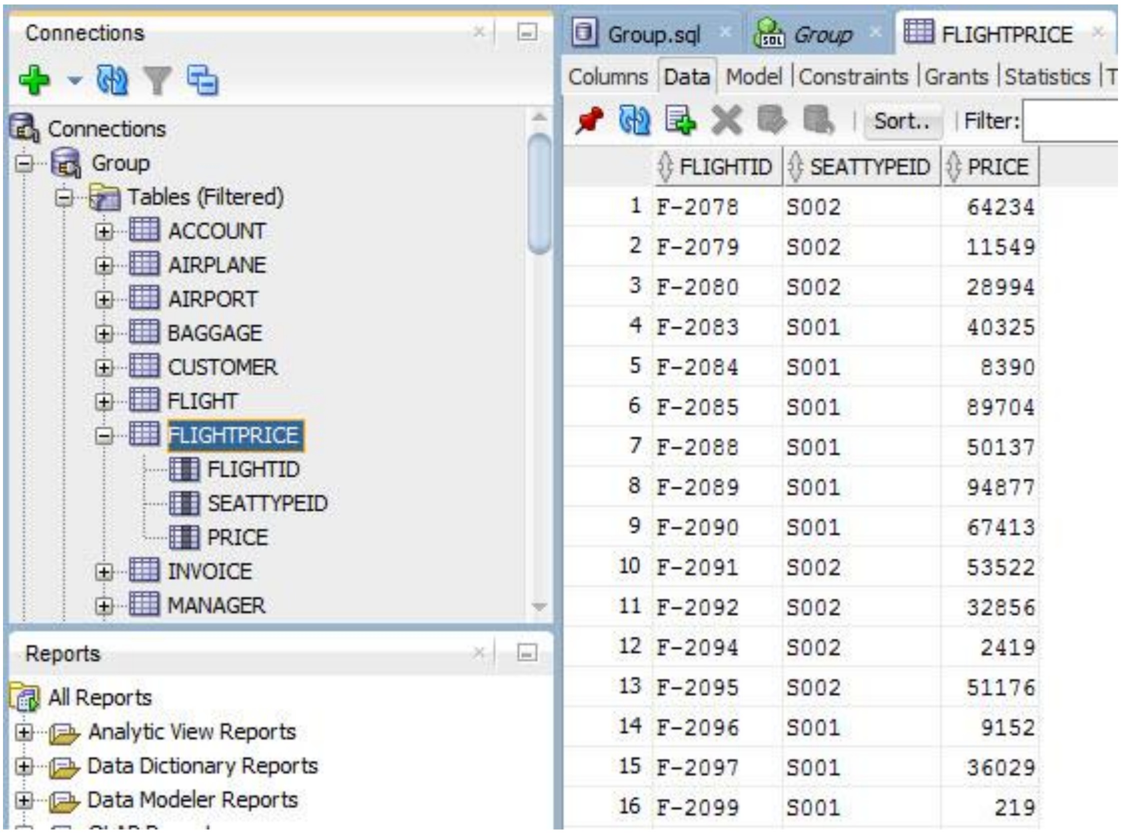


	FLIGHTID	AIRPLANEID	DEPARTDATE	ARRIVALDATE	DEPARTTIME	ARRIVALTIME	DEPARTPORT	ARRIVALPORT	LAYOVER
1	F-1182	HH3536	16-12-16	16-11-16	01-11-17 07:28:00.000000000 PM	01-11-17 10:43:00.000000000 PM	HH88	DH09	1
2	F-1183	DF7774	06-05-17	20-04-17	01-11-17 06:42:00.000000000 PM	01-11-17 05:54:00.000000000 AM	EH3	DH1	1
3	F-1282	HH8499	02-07-17	29-05-17	01-11-17 03:29:00.000000000 PM	01-11-17 05:28:00.000000000 AM	HH9	FF5	0
4	F-1332	HH5200	30-11-16	24-06-17	01-11-17 12:16:00.000000000 AM	01-11-17 04:38:00.000000000 PM	HG3	FH2	0
5	F-1432	FG5114	10-08-17	29-03-17	01-11-17 08:19:00.000000000 PM	01-11-17 10:30:00.000000000 PM	HE3	GF76	1
6	F-1433	GH7187	28-02-17	15-05-17	01-11-17 06:26:00.000000000 PM	01-11-17 12:31:00.000000000 PM	HH8	GF8	1
7	F-1532	HH8827	17-09-17	23-01-17	01-11-17 07:33:00.000000000 AM	01-11-17 01:17:00.000000000 PM	GF5	GH24	0
8	F-1682	HH3740	20-11-16	16-09-17	01-11-17 11:08:00.000000000 AM	01-11-17 10:25:00.000000000 PM	HH3	HF0	1
9	F-1683	GG0893	15-07-17	17-01-17	01-11-17 03:43:00.000000000 PM	01-11-17 03:47:00.000000000 AM	HH47	HF0	0
10	F-1684	HE8284	27-10-17	29-01-17	01-11-17 09:18:00.000000000 AM	01-11-17 08:49:00.000000000 AM	DH1	HF0	0
11	F-1685	CD7679	30-06-17	02-04-17	01-11-17 03:35:00.000000000 AM	01-11-17 07:47:00.000000000 AM	GD2	HF06	0
12	F-1732	HH7009	03-02-17	10-12-16	01-11-17 08:45:00.000000000 AM	01-11-17 09:21:00.000000000 AM	FH7	HF72	0
13	F-1782	HH7996	07-10-17	25-01-17	01-11-17 08:53:00.000000000 AM	01-11-17 08:53:00.000000000 PM	HH7	HG22	1
14	F-1832	HH6842	28-12-16	10-08-17	01-11-17 03:35:00.000000000 AM	01-11-17 08:20:00.000000000 AM	HH8	HG7	1
15	F-1882	HH7718	08-04-17	24-12-16	01-11-17 01:35:00.000000000 AM	01-11-17 08:41:00.000000000 AM	HH3	HH05	1
16	F-1883	GH7337	28-05-17	19-08-17	01-11-17 10:37:00.000000000 AM	01-11-17 06:45:00.000000000 AM	HH6	HH05	1

7. FLIGHTPRICE

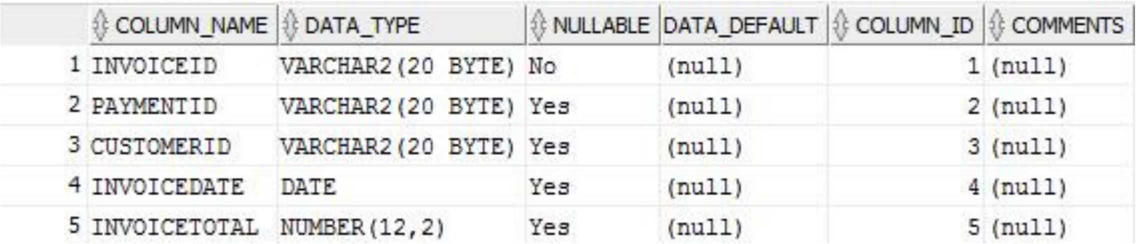


	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	FLIGHTID	VARCHAR2(20 BYTE)	Yes	(null)	1	(null)
2	SEATTYPEID	VARCHAR2(20 BYTE)	Yes	(null)	2	(null)
3	PRICE	NUMBER(10,0)	Yes	(null)	3	(null)

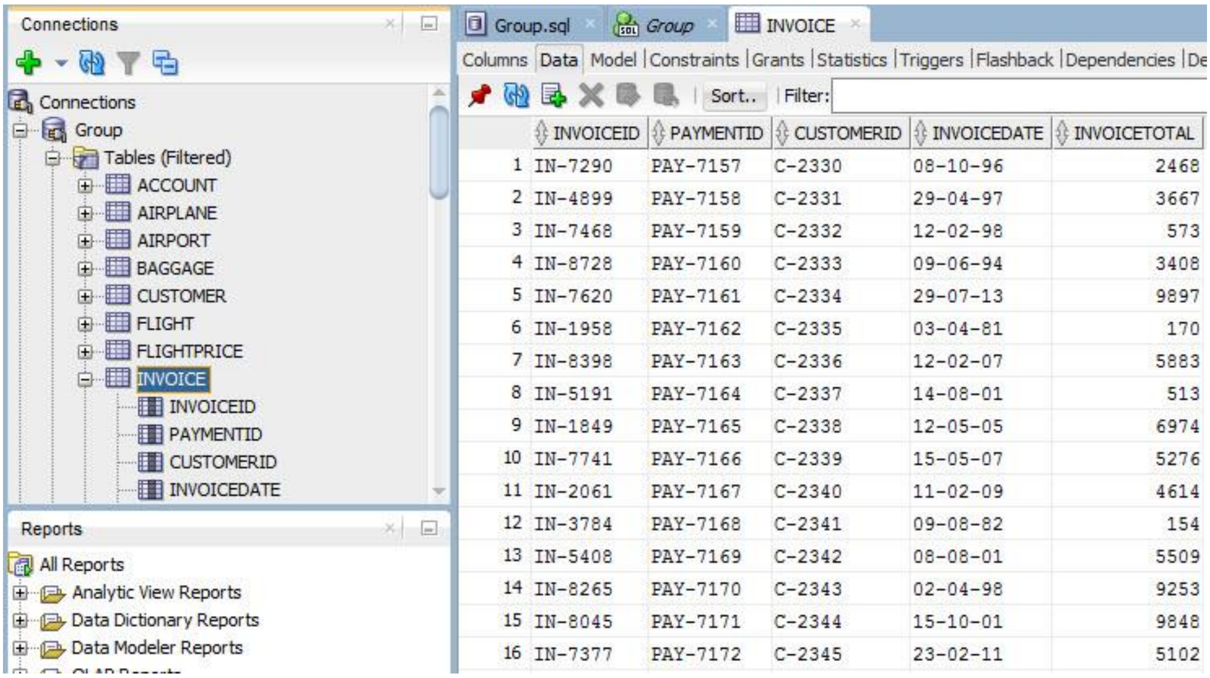


	FLIGHTID	SEATTYPEID	PRICE
1	F-2078	S002	64234
2	F-2079	S002	11549
3	F-2080	S002	28994
4	F-2083	S001	40325
5	F-2084	S001	8390
6	F-2085	S001	89704
7	F-2088	S001	50137
8	F-2089	S001	94877
9	F-2090	S001	67413
10	F-2091	S002	53522
11	F-2092	S002	32856
12	F-2094	S002	2419
13	F-2095	S002	51176
14	F-2096	S001	9152
15	F-2097	S001	36029
16	F-2099	S001	219

8. INVOICE

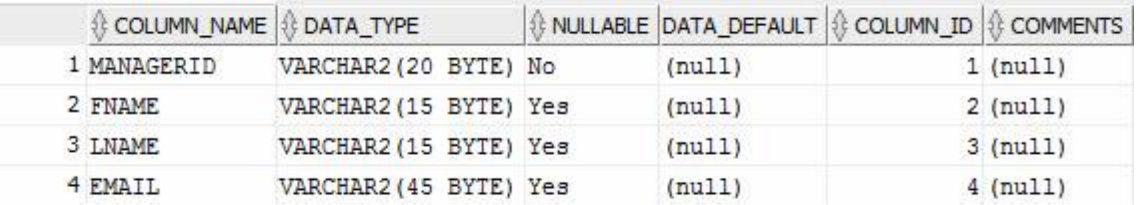


	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	INVOICEID	VARCHAR2(20 BYTE)	No	(null)	1	(null)
2	PAYMENTID	VARCHAR2(20 BYTE)	Yes	(null)	2	(null)
3	CUSTOMERID	VARCHAR2(20 BYTE)	Yes	(null)	3	(null)
4	INVOICEDATE	DATE	Yes	(null)	4	(null)
5	INVOICETOTAL	NUMBER(12,2)	Yes	(null)	5	(null)

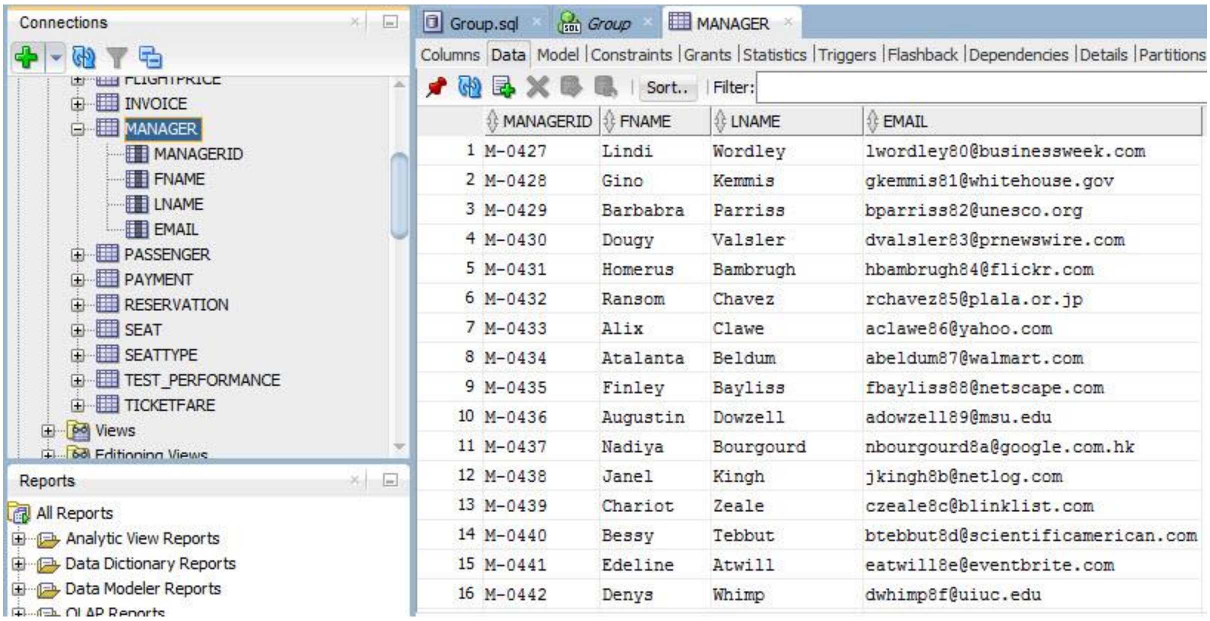


	INVOICEID	PAYMENTID	CUSTOMERID	INVOICEDATE	INVOICETOTAL
1	IN-7290	PAY-7157	C-2330	08-10-96	2468
2	IN-4899	PAY-7158	C-2331	29-04-97	3667
3	IN-7468	PAY-7159	C-2332	12-02-98	573
4	IN-8728	PAY-7160	C-2333	09-06-94	3408
5	IN-7620	PAY-7161	C-2334	29-07-13	9897
6	IN-1958	PAY-7162	C-2335	03-04-81	170
7	IN-8398	PAY-7163	C-2336	12-02-07	5883
8	IN-5191	PAY-7164	C-2337	14-08-01	513
9	IN-1849	PAY-7165	C-2338	12-05-05	6974
10	IN-7741	PAY-7166	C-2339	15-05-07	5276
11	IN-2061	PAY-7167	C-2340	11-02-09	4614
12	IN-3784	PAY-7168	C-2341	09-08-82	154
13	IN-5408	PAY-7169	C-2342	08-08-01	5509
14	IN-8265	PAY-7170	C-2343	02-04-98	9253
15	IN-8045	PAY-7171	C-2344	15-10-01	9848
16	IN-7377	PAY-7172	C-2345	23-02-11	5102

9. MANAGER

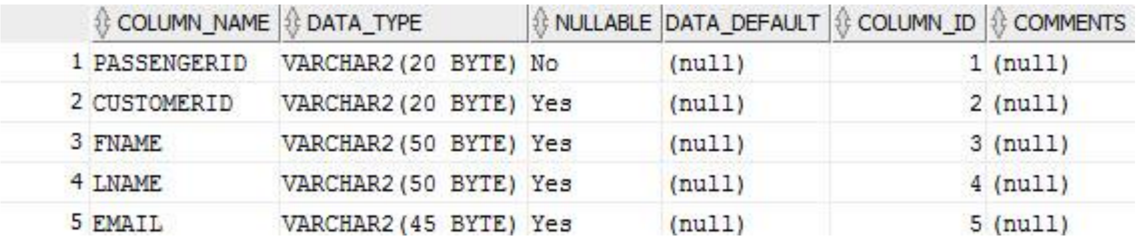


	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	MANAGERID	VARCHAR2(20 BYTE)	No	(null)	1	(null)
2	FNAME	VARCHAR2(15 BYTE)	Yes	(null)	2	(null)
3	LNAME	VARCHAR2(15 BYTE)	Yes	(null)	3	(null)
4	EMAIL	VARCHAR2(45 BYTE)	Yes	(null)	4	(null)

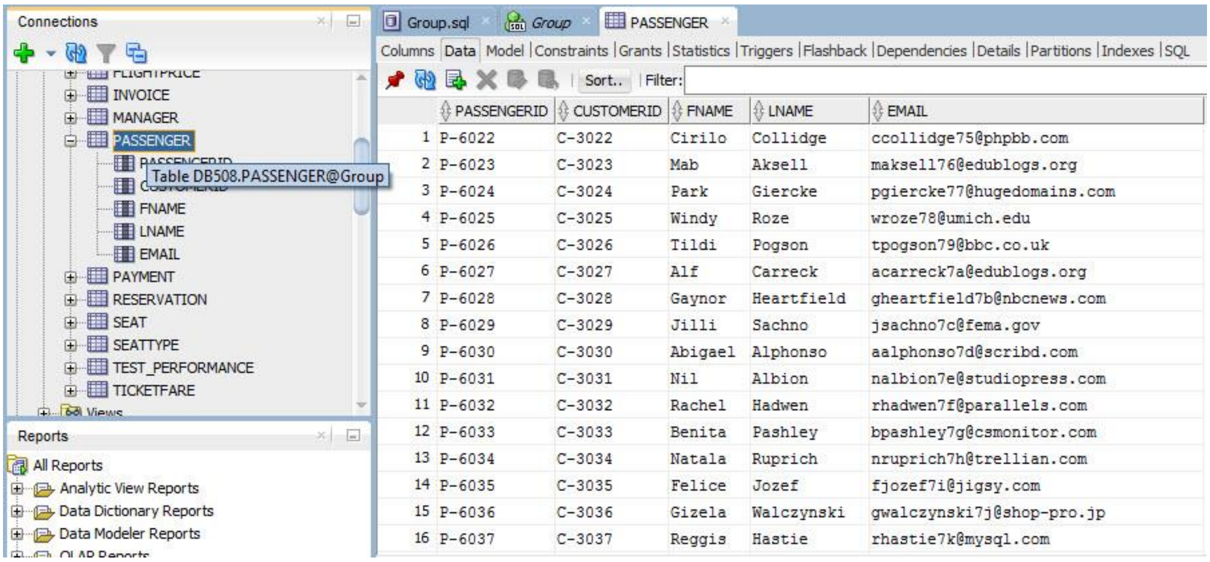


	MANAGERID	FNAME	LNAME	EMAIL
1	M-0427	Lindi	Wordley	lwordley80@businessweek.com
2	M-0428	Gino	Kemmis	gkemmis81@whitehouse.gov
3	M-0429	Barbabra	Parriss	bparriss82@unesco.org
4	M-0430	Dougy	Valsler	dvalsler83@prnewswire.com
5	M-0431	Homerus	Bambrugh	hbambrugh84@flickr.com
6	M-0432	Ransom	Chavez	rchavez85@plala.or.jp
7	M-0433	Alix	Clawe	aclawe86@yahoo.com
8	M-0434	Atalanta	Beldum	abeldum87@walmart.com
9	M-0435	Finley	Bayliss	fbayliss88@netscape.com
10	M-0436	Augustin	Dowzell	adowzell89@msu.edu
11	M-0437	Nadiya	Bourgourd	nbourgourd8a@google.com.hk
12	M-0438	Janel	Kingh	jkingh8b@netlog.com
13	M-0439	Chariot	Zeale	czeale8c@blinklist.com
14	M-0440	Bessy	Tebbut	btebbut8d@scientificamerican.com
15	M-0441	Edeline	Atwill	eatwill8e@eventbrite.com
16	M-0442	Denys	Whimp	dwhimp8f@uiuc.edu

10. PASSENGER

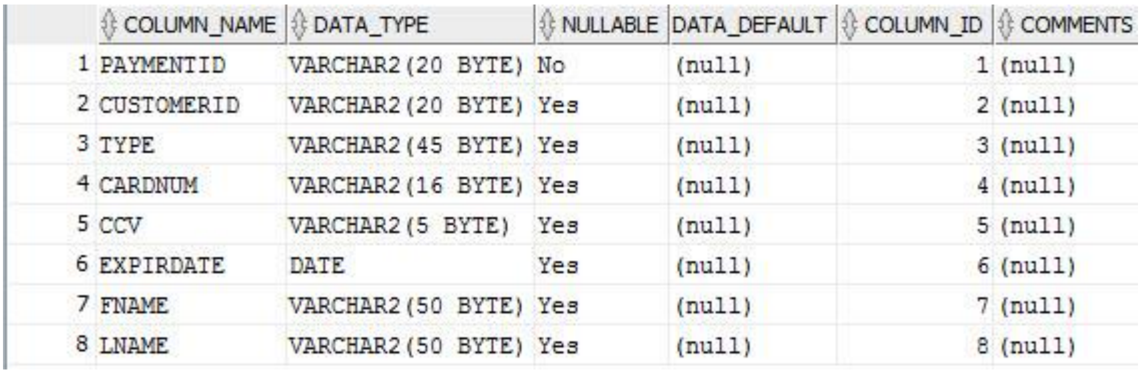


	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	PASSENGERID	VARCHAR2(20 BYTE)	No	(null)	1	(null)
2	CUSTOMERID	VARCHAR2(20 BYTE)	Yes	(null)	2	(null)
3	FNAME	VARCHAR2(50 BYTE)	Yes	(null)	3	(null)
4	LNAME	VARCHAR2(50 BYTE)	Yes	(null)	4	(null)
5	EMAIL	VARCHAR2(45 BYTE)	Yes	(null)	5	(null)

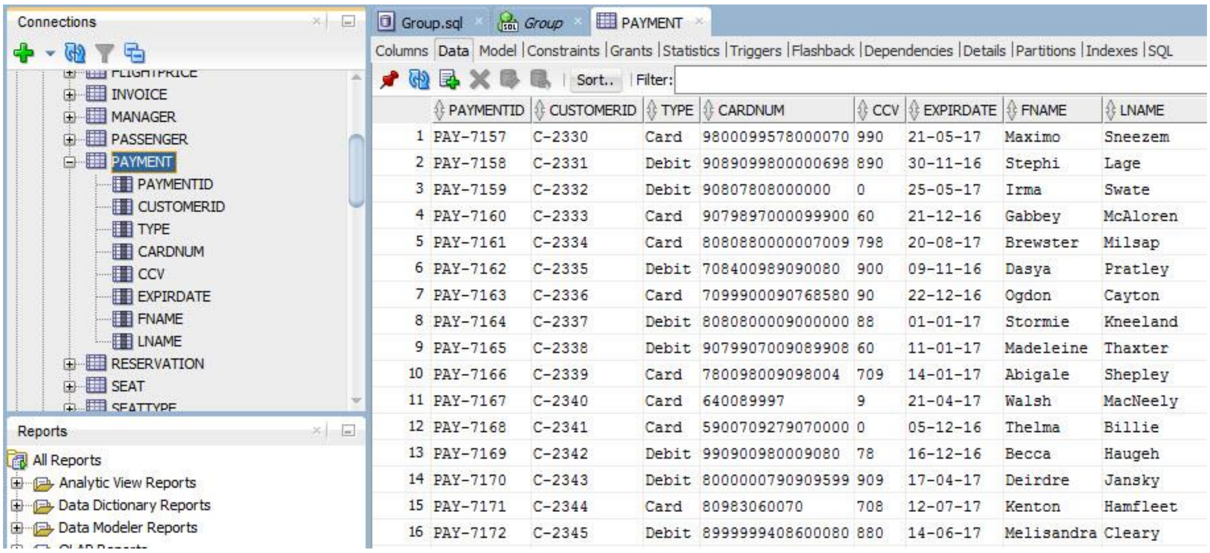


	PASSENGERID	CUSTOMERID	FNAME	LNAME	EMAIL
1	P-6022	C-3022	Cirilo	Collidge	ccollidge75@phpbb.com
2	P-6023	C-3023	Mab	Aksell	maksell76@edublogs.org
3	P-6024	C-3024	Park	Giercke	pgiercke77@hugedomains.com
4	P-6025	C-3025	Windy	Roze	wroze78@umich.edu
5	P-6026	C-3026	Tildi	Pogson	tpogson79@bbc.co.uk
6	P-6027	C-3027	Alf	Carreck	acarreck7a@edublogs.org
7	P-6028	C-3028	Gaynor	Heartfield	gheartfield7b@nbcnews.com
8	P-6029	C-3029	Jilli	Sachno	jsachno7c@fema.gov
9	P-6030	C-3030	Abigael	Alphonso	aalphonso7d@scribd.com
10	P-6031	C-3031	Nil	Albion	nalbion7e@studiopress.com
11	P-6032	C-3032	Rachel	Hadwen	rhadwen7f@parallels.com
12	P-6033	C-3033	Benita	Pashley	bpashley7g@csmonitor.com
13	P-6034	C-3034	Natala	Ruprich	nruprich7h@trellian.com
14	P-6035	C-3035	Felice	Jozef	fjozef7i@jigsy.com
15	P-6036	C-3036	Gizela	Walczynski	gwalczynski7j@shop-pro.jp
16	P-6037	C-3037	Reggis	Hastie	rhastie7k@mysql.com

11. PAYMENT

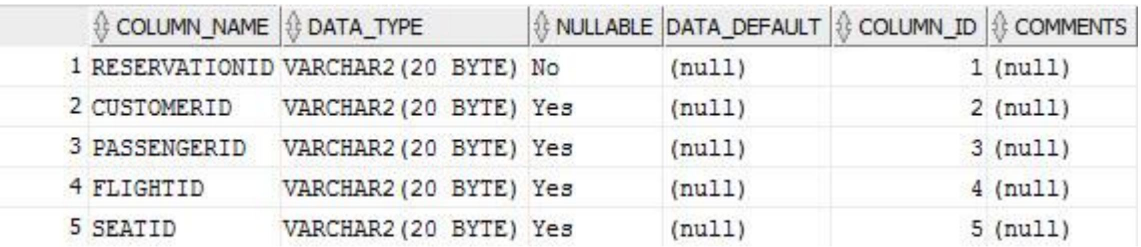


	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	PAYMENTID	VARCHAR2(20 BYTE)	No	(null)	1	(null)
2	CUSTOMERID	VARCHAR2(20 BYTE)	Yes	(null)	2	(null)
3	TYPE	VARCHAR2(45 BYTE)	Yes	(null)	3	(null)
4	CARDNUM	VARCHAR2(16 BYTE)	Yes	(null)	4	(null)
5	CCV	VARCHAR2(5 BYTE)	Yes	(null)	5	(null)
6	EXPIRDATE	DATE	Yes	(null)	6	(null)
7	FNAME	VARCHAR2(50 BYTE)	Yes	(null)	7	(null)
8	LNAME	VARCHAR2(50 BYTE)	Yes	(null)	8	(null)

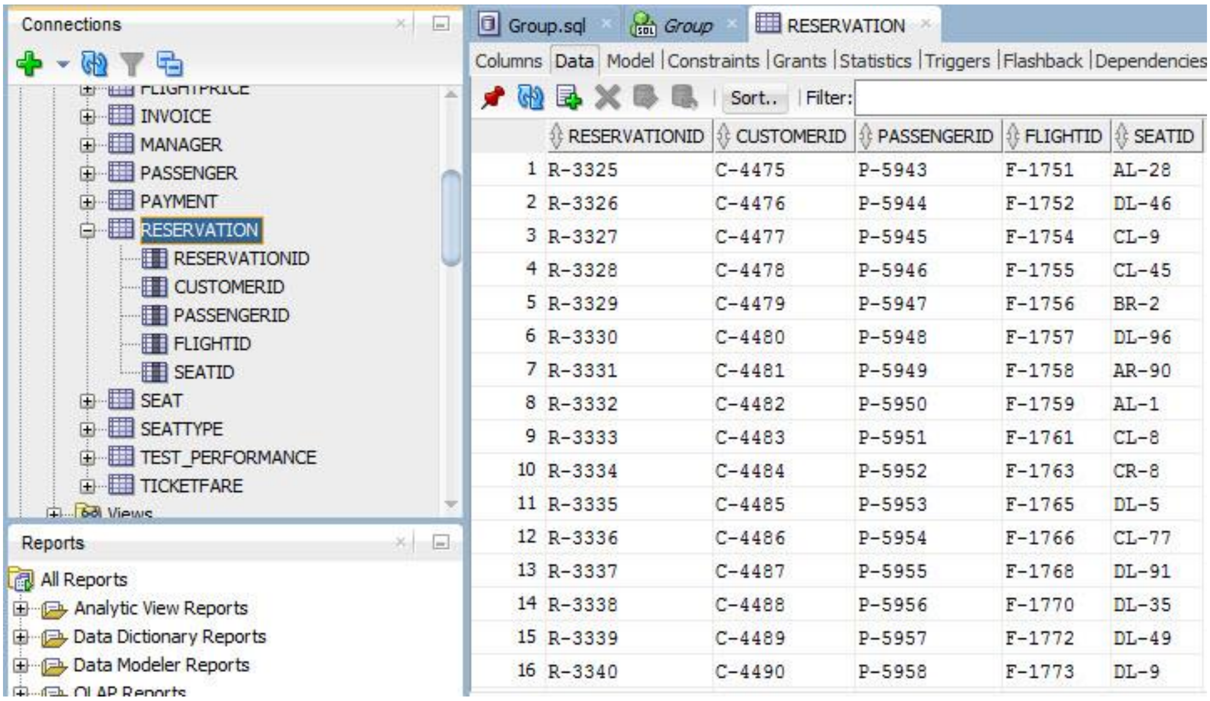


	PAYMENTID	CUSTOMERID	TYPE	CARDNUM	CCV	EXPIRDATE	FNAME	LNAME
1	PAY-7157	C-2330	Card	9800099578000070	990	21-05-17	Maximo	Sneezem
2	PAY-7158	C-2331	Debit	9089099800000698	890	30-11-16	Stephi	Lage
3	PAY-7159	C-2332	Debit	90807808000000	0	25-05-17	Irma	Swate
4	PAY-7160	C-2333	Card	9079897000099900	60	21-12-16	Gabbey	McAloren
5	PAY-7161	C-2334	Card	8080880000007009	798	20-08-17	Brewster	Milsap
6	PAY-7162	C-2335	Debit	708400989090080	900	09-11-16	Dasya	Pratley
7	PAY-7163	C-2336	Card	7099900090768580	90	22-12-16	Ogdon	Cayton
8	PAY-7164	C-2337	Debit	8080800009000000	88	01-01-17	Stormie	Kneeland
9	PAY-7165	C-2338	Debit	9079907009089908	60	11-01-17	Madeleine	Thaxter
10	PAY-7166	C-2339	Card	780098009098004	709	14-01-17	Abigale	Shepley
11	PAY-7167	C-2340	Card	640089997	9	21-04-17	Walsh	MacNeely
12	PAY-7168	C-2341	Card	5900709279070000	0	05-12-16	Thelma	Billie
13	PAY-7169	C-2342	Debit	990900980009080	78	16-12-16	Becca	Haugeh
14	PAY-7170	C-2343	Debit	8000000790909599	909	17-04-17	Deirdre	Jansky
15	PAY-7171	C-2344	Card	80983060070	708	12-07-17	Kenton	Hamfleet
16	PAY-7172	C-2345	Debit	8999999408600080	880	14-06-17	Melisandra	Cleary

12. RESERVATION

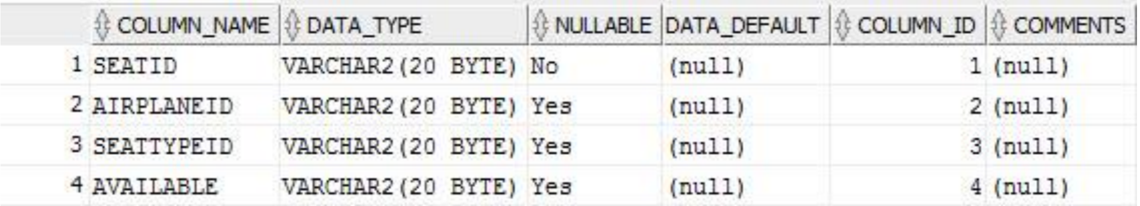


	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	RESERVATIONID	VARCHAR2(20 BYTE)	No	(null)	1	(null)
2	CUSTOMERID	VARCHAR2(20 BYTE)	Yes	(null)	2	(null)
3	PASSENGERID	VARCHAR2(20 BYTE)	Yes	(null)	3	(null)
4	FLIGHTID	VARCHAR2(20 BYTE)	Yes	(null)	4	(null)
5	SEATID	VARCHAR2(20 BYTE)	Yes	(null)	5	(null)

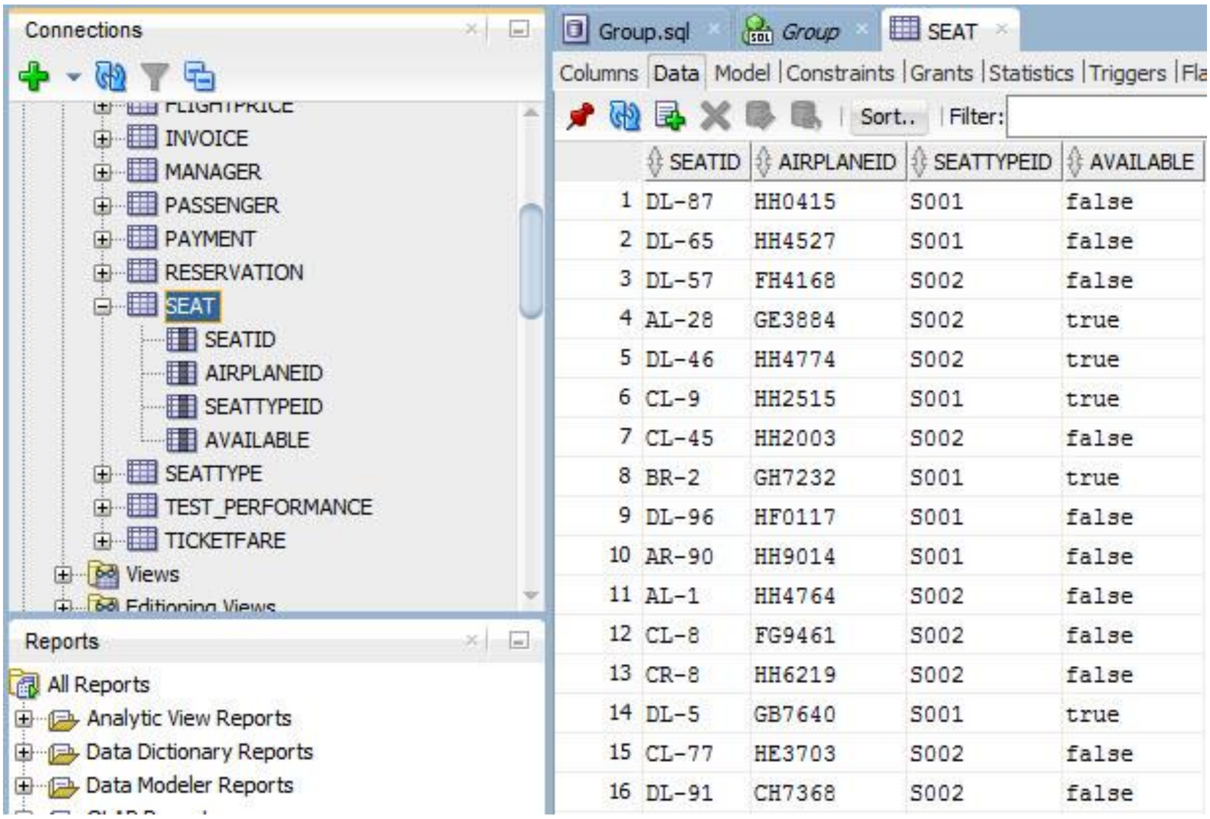


	RESERVATIONID	CUSTOMERID	PASSENGERID	FLIGHTID	SEATID
1	R-3325	C-4475	P-5943	F-1751	AL-28
2	R-3326	C-4476	P-5944	F-1752	DL-46
3	R-3327	C-4477	P-5945	F-1754	CL-9
4	R-3328	C-4478	P-5946	F-1755	CL-45
5	R-3329	C-4479	P-5947	F-1756	BR-2
6	R-3330	C-4480	P-5948	F-1757	DL-96
7	R-3331	C-4481	P-5949	F-1758	AR-90
8	R-3332	C-4482	P-5950	F-1759	AL-1
9	R-3333	C-4483	P-5951	F-1761	CL-8
10	R-3334	C-4484	P-5952	F-1763	CR-8
11	R-3335	C-4485	P-5953	F-1765	DL-5
12	R-3336	C-4486	P-5954	F-1766	CL-77
13	R-3337	C-4487	P-5955	F-1768	DL-91
14	R-3338	C-4488	P-5956	F-1770	DL-35
15	R-3339	C-4489	P-5957	F-1772	DL-49
16	R-3340	C-4490	P-5958	F-1773	DL-9

13. SEAT

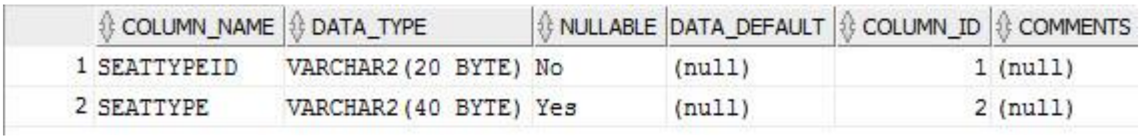


	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	SEATID	VARCHAR2(20 BYTE)	No	(null)	1	(null)
2	AIRPLANEID	VARCHAR2(20 BYTE)	Yes	(null)	2	(null)
3	SEATTYPEID	VARCHAR2(20 BYTE)	Yes	(null)	3	(null)
4	AVAILABLE	VARCHAR2(20 BYTE)	Yes	(null)	4	(null)

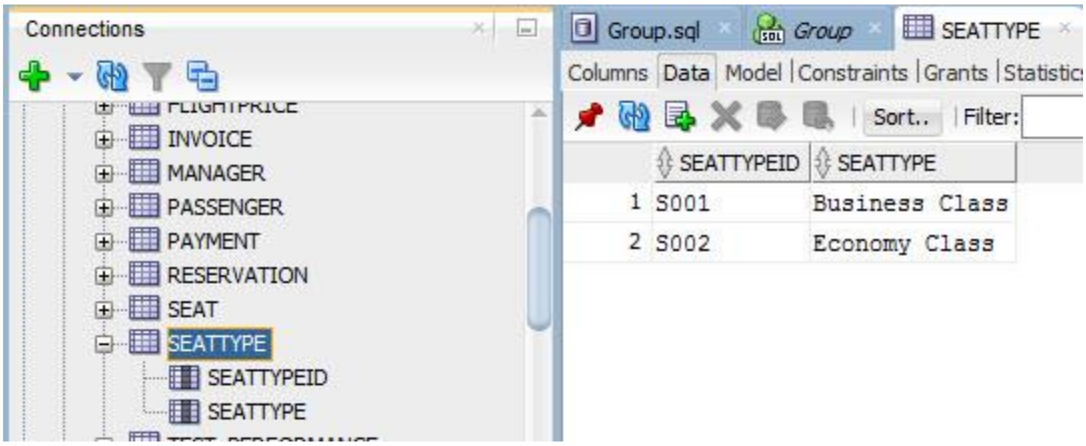


	SEATID	AIRPLANEID	SEATTYPEID	AVAILABLE
1	DL-87	HH0415	S001	false
2	DL-65	HH4527	S001	false
3	DL-57	FH4168	S002	false
4	AL-28	GE3884	S002	true
5	DL-46	HH4774	S002	true
6	CL-9	HH2515	S001	true
7	CL-45	HH2003	S002	false
8	BR-2	GH7232	S001	true
9	DL-96	HF0117	S001	false
10	AR-90	HH9014	S001	false
11	AL-1	HH4764	S002	false
12	CL-8	FG9461	S002	false
13	CR-8	HH6219	S002	false
14	DL-5	GB7640	S001	true
15	CL-77	HE3703	S002	false
16	DL-91	CH7368	S002	false

14. SEAT-TYPE

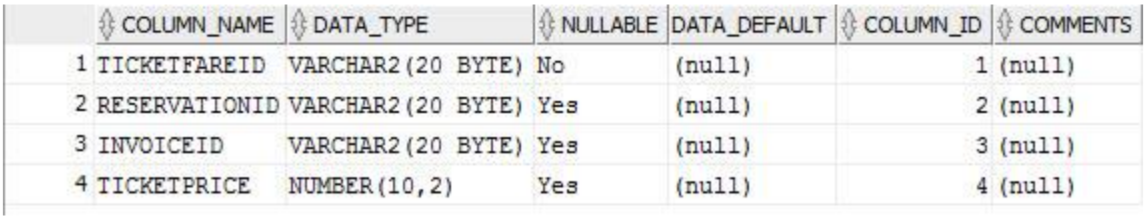


	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	SEATTYPEID	VARCHAR2(20 BYTE)	No	(null)	1	(null)
2	SEATTYPE	VARCHAR2(40 BYTE)	Yes	(null)	2	(null)

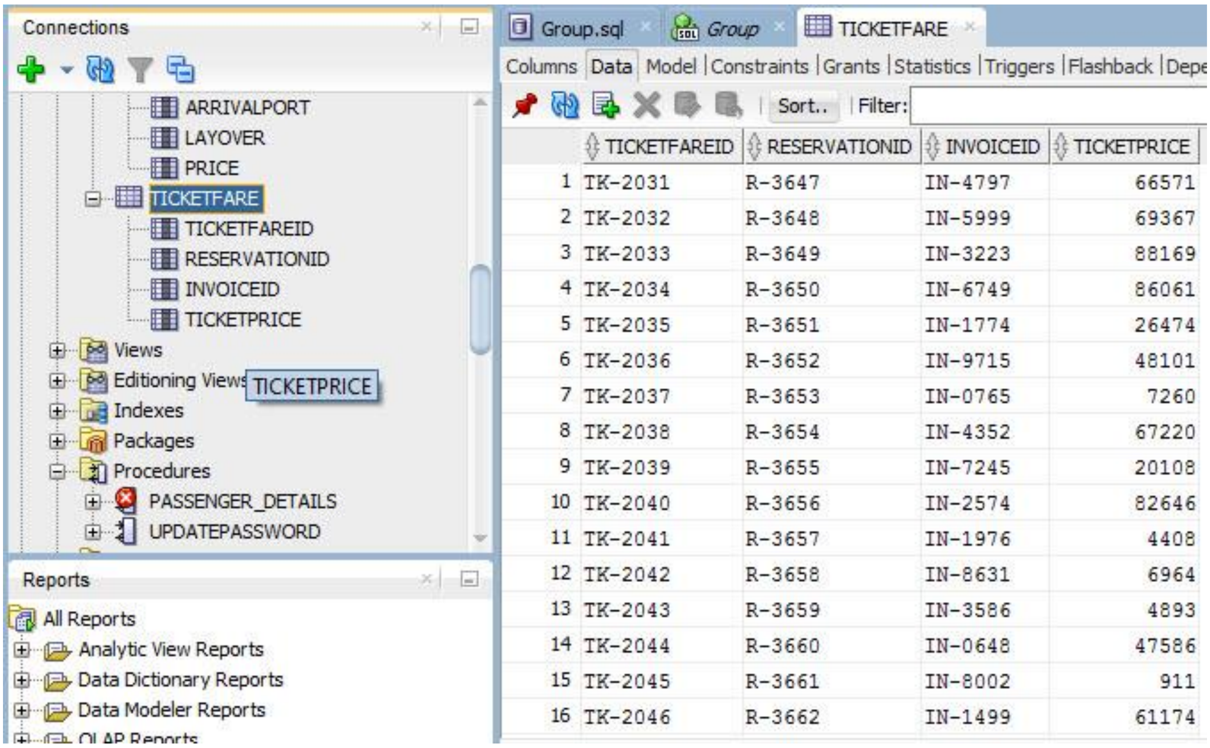


	SEATTYPEID	SEATTYPE
1	S001	Business Class
2	S002	Economy Class

15. TICKETFARE



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	TICKETFAREID	VARCHAR2(20 BYTE)	No	(null)	1	(null)
2	RESERVATIONID	VARCHAR2(20 BYTE)	Yes	(null)	2	(null)
3	INVOICEID	VARCHAR2(20 BYTE)	Yes	(null)	3	(null)
4	TICKETPRICE	NUMBER(10,2)	Yes	(null)	4	(null)



	TICKETFAREID	RESERVATIONID	INVOICEID	TICKETPRICE
1	TK-2031	R-3647	IN-4797	66571
2	TK-2032	R-3648	IN-5999	69367
3	TK-2033	R-3649	IN-3223	88169
4	TK-2034	R-3650	IN-6749	86061
5	TK-2035	R-3651	IN-1774	26474
6	TK-2036	R-3652	IN-9715	48101
7	TK-2037	R-3653	IN-0765	7260
8	TK-2038	R-3654	IN-4352	67220
9	TK-2039	R-3655	IN-7245	20108
10	TK-2040	R-3656	IN-2574	82646
11	TK-2041	R-3657	IN-1976	4408
12	TK-2042	R-3658	IN-8631	6964
13	TK-2043	R-3659	IN-3586	4893
14	TK-2044	R-3660	IN-0648	47586
15	TK-2045	R-3661	IN-8002	911
16	TK-2046	R-3662	IN-1499	61174

7. Performance Tuning:

SQL statements are used to fetch the data from database. We can write the queries in many ways to fetch the data but writing in a best way to is important when we consider the performance. Below are a few instances

Retrieval becomes faster when we use the actual names of columns of the tables instead id '*'

```
Select * From Flightprice;
```

Retrieval time- 0.21

```
Select Flightid,Seattypeid,Price From Flightprice;
```

Retrieval time-0.134

Always use Union all instead of Union

```
SELECT customerid FROM customer
UNION ALL
SELECT customerid FROM payment;
```

Retrieval time-0.171

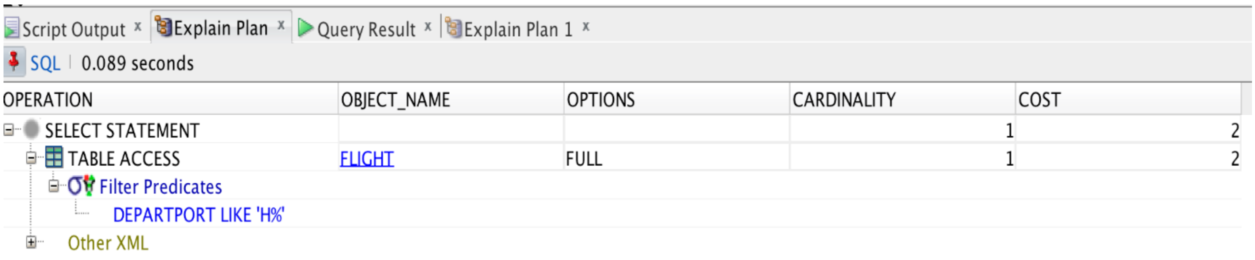
```
SELECT customerid FROM customer
UNION
SELECT customerid FROM payment;
```

Retrieval time-0.11

Above are just a few tricks we could use while running simple queries to improve performance. Apart from these we could some additional features like indexing, hints, views etc to improve performance.

INDEXING

```
SELECT  FLIGHTID,DEPARTDATE,ARRIVALDATE,ARRIVALPORT FROM FLIGHT WHERE
DEPARTPORT LIKE 'H%';
```

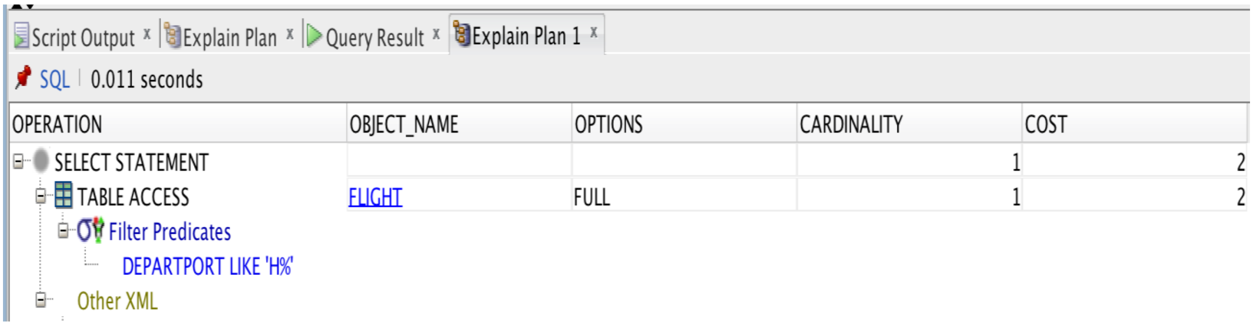


0.089 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	2
TABLE ACCESS	FLIGHT	FULL	1	2
Filter Predicates				
DEPARTPORT LIKE 'H%'				
Other XML				

```
CREATE INDEX FLIGHT_IDX ON FLIGHT (FLIGHTID,DEPARTPORT, ARRIVALPORT);
```

```
SELECT  FLIGHTID,DEPARTDATE,ARRIVALDATE,ARRIVALPORT FROM FLIGHT WHERE
DEPARTPORT LIKE 'H%';
```



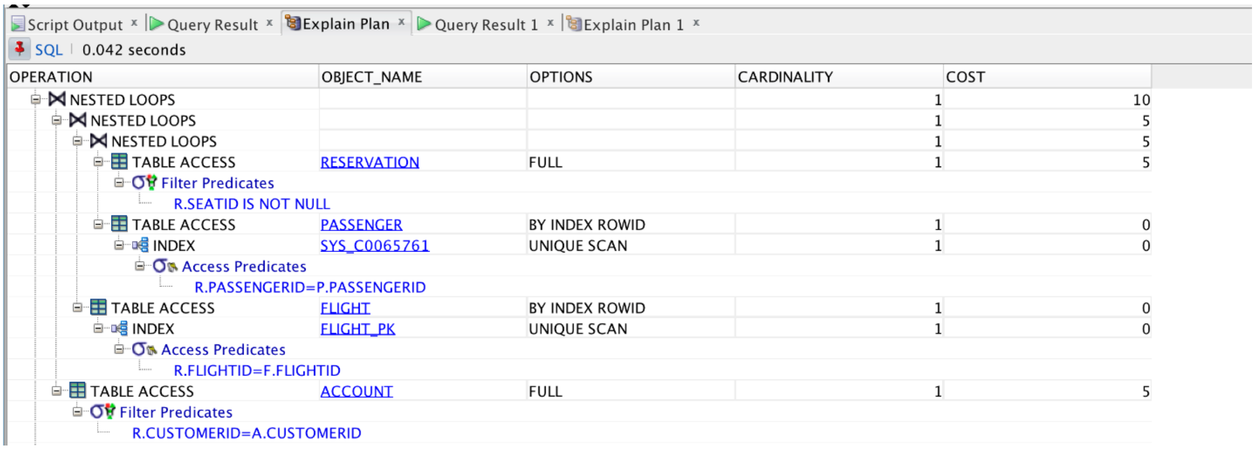
SQL | 0.011 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	2
TABLE ACCESS	FLIGHT	FULL	1	2
Filter Predicates				
DEPARTPORT LIKE 'H%'				
Other XML				

As we can notice, the retrieval time for the result set has decreased drastically after the use of indexes.

OPTIMIZATION

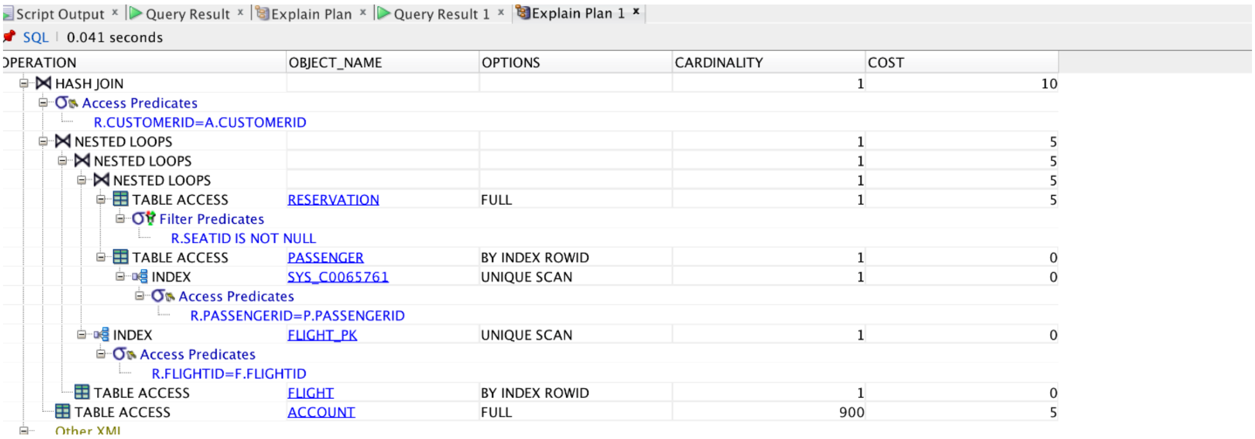
```
ALTER SESSION SET OPTIMIZER_MODE = FIRST_ROWS;
```



SQL | 0.042 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
NESTED LOOPS			1	10
NESTED LOOPS			1	5
NESTED LOOPS			1	5
TABLE ACCESS	RESERVATION	FULL	1	5
Filter Predicates				
R.SEATID IS NOT NULL				
TABLE ACCESS	PASSENGER	BY INDEX ROWID	1	0
INDEX	SYS_C0065761	UNIQUE SCAN	1	0
Access Predicates				
R.PASSENGERID=P.PASSENGERID				
TABLE ACCESS	FLIGHT	BY INDEX ROWID	1	0
INDEX	FLIGHT_PK	UNIQUE SCAN	1	0
Access Predicates				
R.FLIGHTID=F.FLIGHTID				
TABLE ACCESS	ACCOUNT	FULL	1	5
Filter Predicates				
R.CUSTOMERID=A.CUSTOMERID				

```
ALTER SESSION SET OPTIMIZER_MODE = ALL_ROWS;
```



SQL | 0.041 seconds

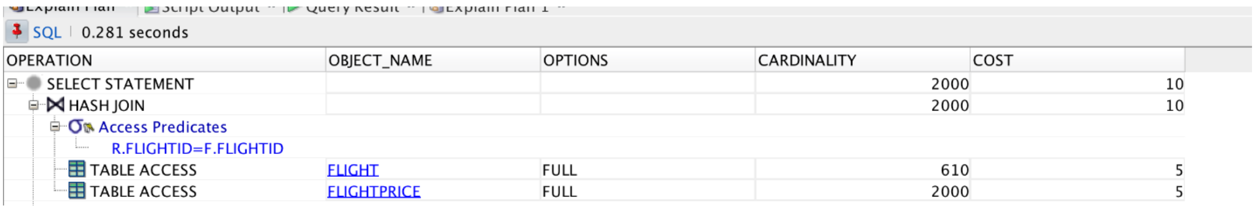
OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
HASH JOIN			1	10
Access Predicates				
R.CUSTOMERID=A.CUSTOMERID				
NESTED LOOPS			1	5
NESTED LOOPS			1	5
NESTED LOOPS			1	5
TABLE ACCESS	RESERVATION	FULL	1	5
Filter Predicates				
R.SEATID IS NOT NULL				
TABLE ACCESS	PASSENGER	BY INDEX ROWID	1	0
INDEX	SYS_C0065761	UNIQUE SCAN	1	0
Access Predicates				
R.PASSENGERID=P.PASSENGERID				
INDEX	FLIGHT_PK	UNIQUE SCAN	1	0
Access Predicates				
R.FLIGHTID=F.FLIGHTID				
TABLE ACCESS	FLIGHT	BY INDEX ROWID	1	0
TABLE ACCESS	ACCOUNT	FULL	900	5
Other XML				

We can see there is reduction in cost by 5 when we alter the session to first_rows instead of all_rows.

MATERIALIZED VIEW

Below query can be used to select data that has to be displayed on the first page of any airline booking website. This is a query that used n number of times, hence creating a materialized view for the same would prove to be very beneficial.

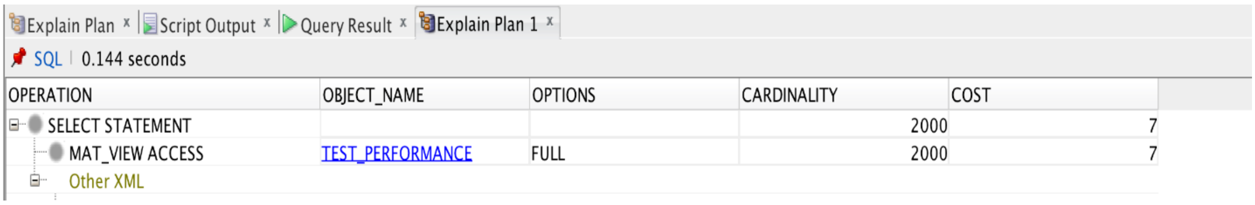
```
SELECT r.flightid,
  r.departdate,
  r.arrivaldate,
  r.departtime,
  r.arrivaltime,
  r.departport,
  r.arrivalport,
  r.layover,
  f.price
FROM flight r
INNER JOIN flightprice f
ON r.flightid = f.flightid;
```



SQL | 0.281 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			2000	10
HASH JOIN			2000	10
Access Predicates				
R.FLIGHTID=F.FLIGHTID				
TABLE ACCESS	FLIGHT	FULL	610	5
TABLE ACCESS	FLIGHTPRICE	FULL	2000	5

```
CREATE MATERIALIZED VIEW test_performance
AS
   (SELECT r.flightid,
     r.departdate,
     r.arrivaldate,
     r.departtime,
     r.arrivaltime,
     r.departport,
     r.arrivalport,
     r.layover,
     f.price
  FROM flight r
  INNER JOIN flightprice f
  ON r.flightid = f.flightid
  );
  SELECT * FROM test_performance;
```



0.144 seconds

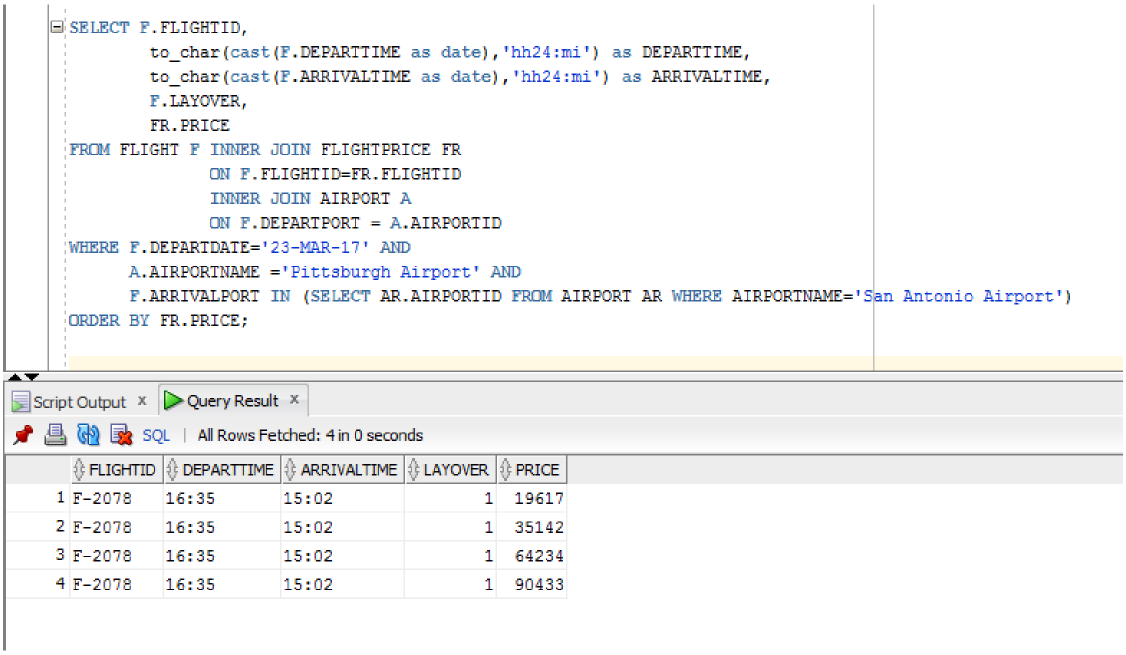
OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			2000	7
MAT_VIEW ACCESS	TEST_PERFORMANCE	FULL	2000	7
Other XML				

8. SQL Example Queries:

Query 1:

Display flights that are available from 'Pittsburgh Airport' to 'San Antonio Airport' on date '23-MAR-17' sorted in ascending order by flight price . Include flight details such as flight ID, departure time, arrival time, departure port, arrival port, layover and price for that flight.

```sql
SELECT F.FLIGHTID,
        to_char(cast(F.DEPARTTIME as date),'hh24:mi') as DEPARTTIME,
        to_char(cast(F.ARRIVALTIME as date),'hh24:mi') as ARRIVALTIME,
        F.LAYOVER,
        FR.PRICE
FROM FLIGHT F INNER JOIN FLIGHTPRICE FR
              ON F.FLIGHTID=FR.FLIGHTID
              INNER JOIN AIRPORT A
              ON F.DEPARTPORT = A.AIRPORTID
WHERE F.DEPARTDATE='23-MAR-17' AND
      A.AIRPORTNAME ='Pittsburgh Airport' AND
      F.ARRIVALPORT IN (SELECT AR.AIRPORTID FROM AIRPORT AR WHERE
AIRPORTNAME='San Antonio Airport')
ORDER BY FR.PRICE;
```



```sql
SELECT F.FLIGHTID,
        to_char(cast(F.DEPARTTIME as date),'hh24:mi') as DEPARTTIME,
        to_char(cast(F.ARRIVALTIME as date),'hh24:mi') as ARRIVALTIME,
        F.LAYOVER,
        FR.PRICE
FROM FLIGHT F INNER JOIN FLIGHTPRICE FR
              ON F.FLIGHTID=FR.FLIGHTID
              INNER JOIN AIRPORT A
              ON F.DEPARTPORT = A.AIRPORTID
WHERE F.DEPARTDATE='23-MAR-17' AND
      A.AIRPORTNAME ='Pittsburgh Airport' AND
      F.ARRIVALPORT IN (SELECT AR.AIRPORTID FROM AIRPORT AR WHERE AIRPORTNAME='San Antonio Airport')
ORDER BY FR.PRICE;
```

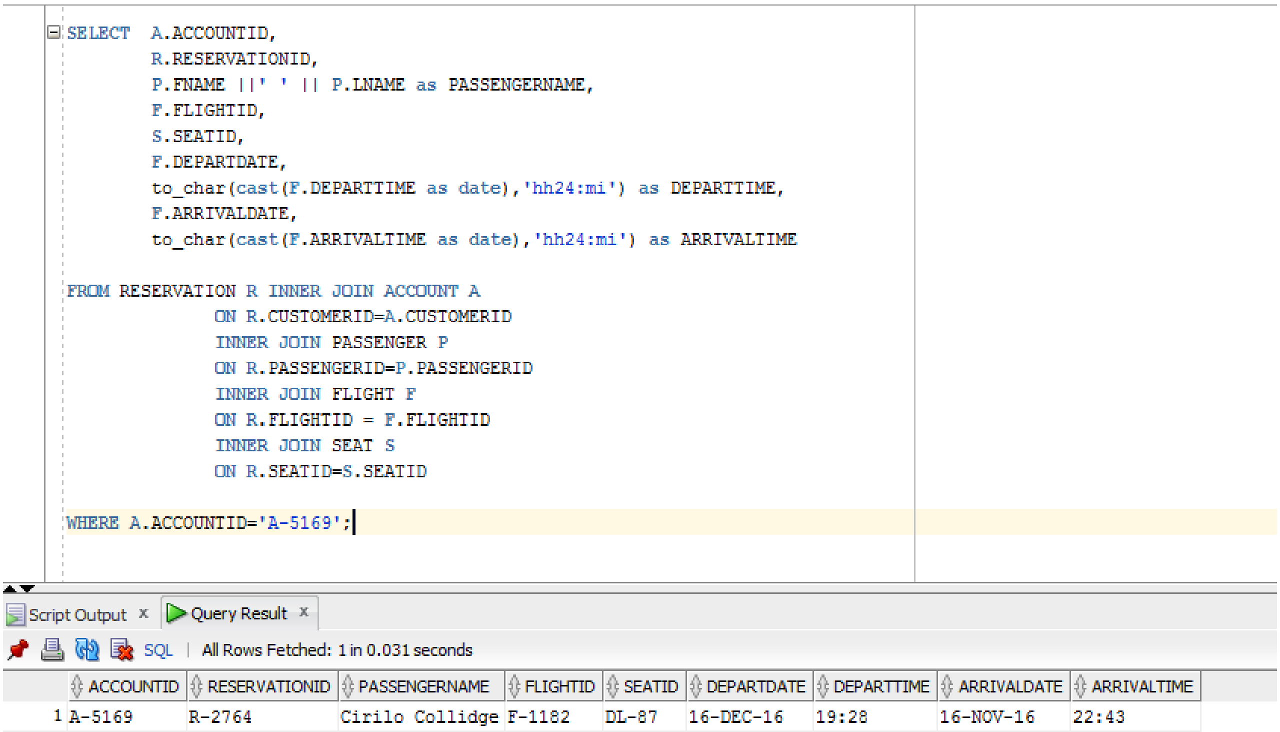
All Rows Fetched: 4 in 0 seconds

```
    FLIGHTID  DEPARTTIME  ARRIVALTIME  LAYOVER  PRICE
  1 F-2078    16:35       15:02              1  19617
  2 F-2078    16:35       15:02              1  35142
  3 F-2078    16:35       15:02              1  64234
  4 F-2078    16:35       15:02              1  90433
```

Query 2:

For a given account id, A-5169 find out the reservations under that account, depicting reservation id, passenger name, flight id, seat id departure date and time and arrival date and time?

```
SELECT  A.ACCOUNTID,
        R.RESERVATIONID,
        P.FNAME ||' ' || P.LNAME as PASSENGERNAME,
        F.FLIGHTID,
        S.SEATID,
        F.DEPARTDATE,
        to_char(cast(F.DEPARTTIME as date),'hh24:mi') as DEPARTTIME,
        F.ARRIVALDATE,
        to_char(cast(F.ARRIVALTIME as date),'hh24:mi') as ARRIVALTIME
FROM RESERVATION R INNER JOIN ACCOUNT A
              ON R.CUSTOMERID=A.CUSTOMERID
              INNER JOIN PASSENGER P
              ON R.PASSENGERID=P.PASSENGERID
              INNER JOIN FLIGHT F
              ON R.FLIGHTID = F.FLIGHTID
              INNER JOIN SEAT S
              ON R.SEATID=S.SEATID
WHERE A.ACCOUNTID='A-5169';
```



```
SELECT  A.ACCOUNTID,
        R.RESERVATIONID,
        P.FNAME ||' ' || P.LNAME as PASSENGERNAME,
        F.FLIGHTID,
        S.SEATID,
        F.DEPARTDATE,
        to_char(cast(F.DEPARTTIME as date),'hh24:mi') as DEPARTTIME,
        F.ARRIVALDATE,
        to_char(cast(F.ARRIVALTIME as date),'hh24:mi') as ARRIVALTIME

FROM RESERVATION R INNER JOIN ACCOUNT A
              ON R.CUSTOMERID=A.CUSTOMERID
              INNER JOIN PASSENGER P
              ON R.PASSENGERID=P.PASSENGERID
              INNER JOIN FLIGHT F
              ON R.FLIGHTID = F.FLIGHTID
              INNER JOIN SEAT S
              ON R.SEATID=S.SEATID

WHERE A.ACCOUNTID='A-5169';
```

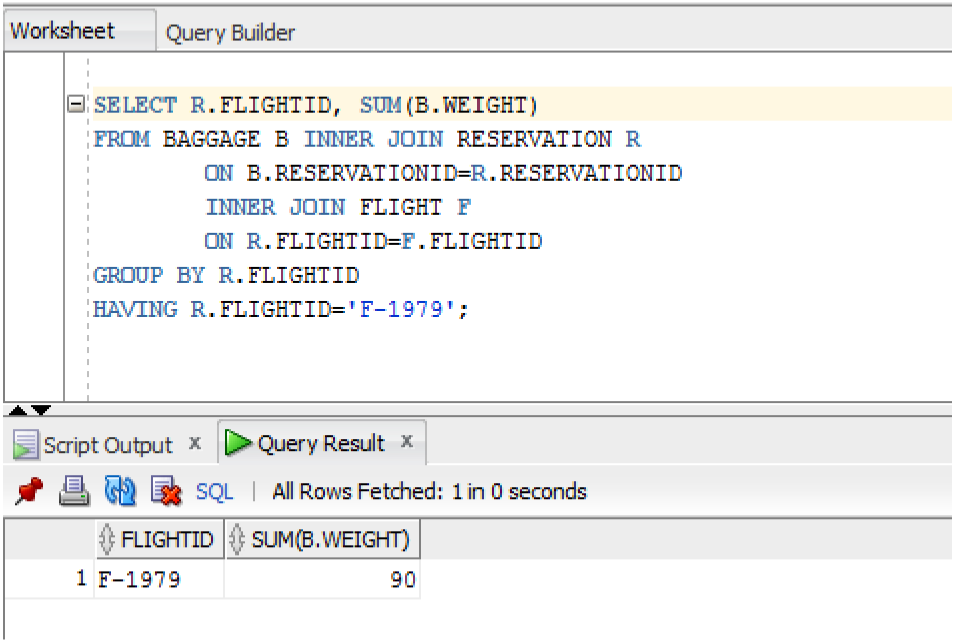
All Rows Fetched: 1 in 0.031 seconds

	ACCOUNTID	RESERVATIONID	PASSENGERNAME	FLIGHTID	SEATID	DEPARTDATE	DEPARTTIME	ARRIVALDATE	ARRIVALTIME
1	A-5169	R-2764	Cirilo Collidge	F-1182	DL-87	16-DEC-16	19:28	16-NOV-16	22:43

Query 3:

What is the total baggage weight given flight with ID 'F-1979' is carrying.

```
SELECT R.FLIGHTID, SUM(B.WEIGHT)
FROM BAGGAGE B INNER JOIN RESERVATION R
        ON B.RESERVATIONID=R.RESERVATIONID
        INNER JOIN FLIGHT F
        ON R.FLIGHTID=F.FLIGHTID
GROUP BY R.FLIGHTID
HAVING R.FLIGHTID='F-1979';
```



```
SELECT R.FLIGHTID, SUM(B.WEIGHT)
FROM BAGGAGE B INNER JOIN RESERVATION R
        ON B.RESERVATIONID=R.RESERVATIONID
        INNER JOIN FLIGHT F
        ON R.FLIGHTID=F.FLIGHTID
GROUP BY R.FLIGHTID
HAVING R.FLIGHTID='F-1979';
```

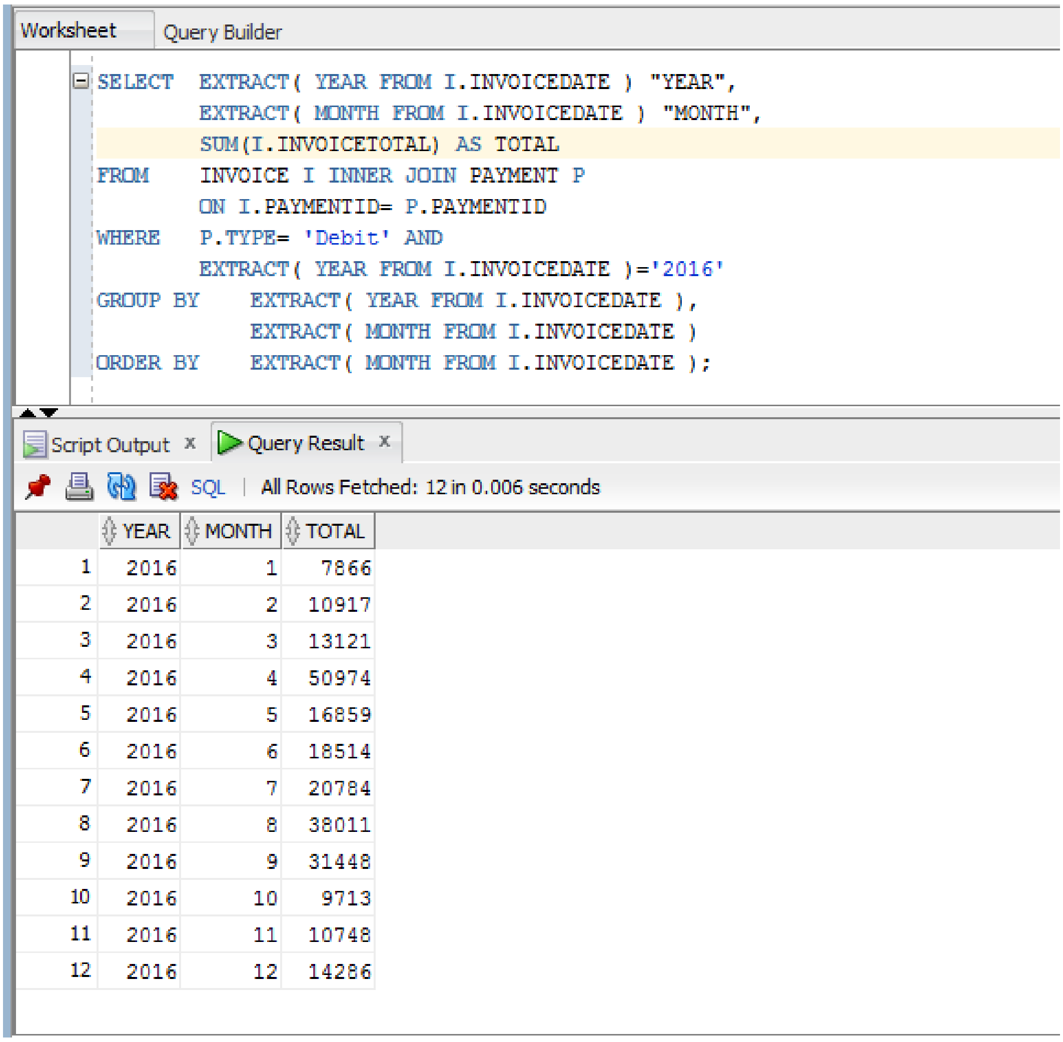
All Rows Fetched: 1 in 0 seconds

	FLIGHTID	SUM(B.WEIGHT)
1	F-1979	90

Query 4:

Query to show Total amount received through Debit Card month wise for a year 2016.

```
SELECT  EXTRACT( YEAR FROM I.INVOICEDATE ) "YEAR",
    EXTRACT( MONTH FROM I.INVOICEDATE ) "MONTH",
    SUM(I.INVOICETOTAL) AS TOTAL
FROM INVOICE I INNER JOIN PAYMENT P
    ON I.PAYMENTID= P.PAYMENTID
WHERE P.TYPE= 'Debit' AND
      EXTRACT( YEAR FROM I.INVOICEDATE )='2016'
GROUP BY EXTRACT( YEAR FROM I.INVOICEDATE ),
         EXTRACT( MONTH FROM I.INVOICEDATE )
ORDER BY EXTRACT( MONTH FROM I.INVOICEDATE );
```



```
SELECT  EXTRACT( YEAR FROM I.INVOICEDATE ) "YEAR",
        EXTRACT( MONTH FROM I.INVOICEDATE ) "MONTH",
        SUM(I.INVOICETOTAL) AS TOTAL
FROM    INVOICE I INNER JOIN PAYMENT P
        ON I.PAYMENTID= P.PAYMENTID
WHERE   P.TYPE= 'Debit' AND
        EXTRACT( YEAR FROM I.INVOICEDATE )='2016'
GROUP BY    EXTRACT( YEAR FROM I.INVOICEDATE ),
            EXTRACT( MONTH FROM I.INVOICEDATE )
ORDER BY    EXTRACT( MONTH FROM I.INVOICEDATE );
```

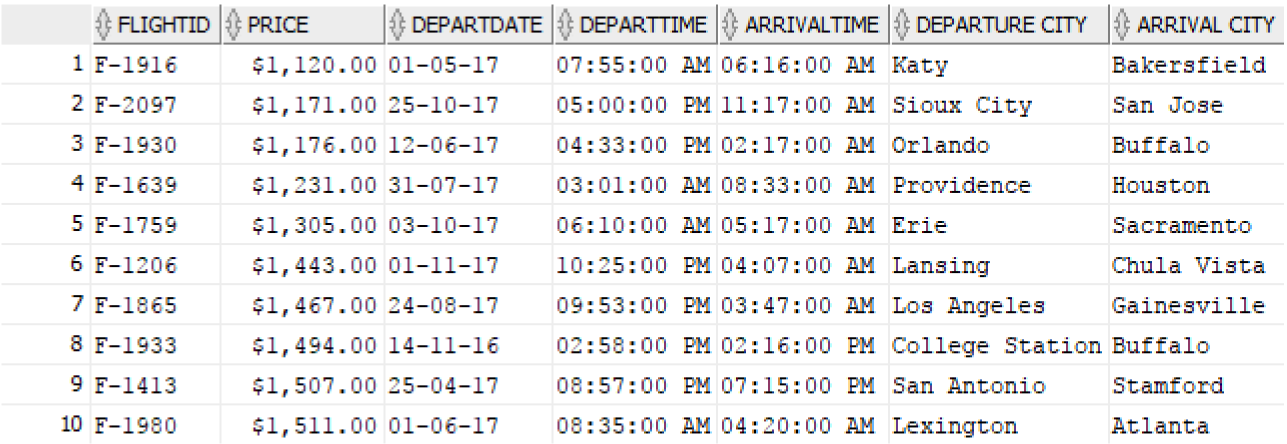
All Rows Fetched: 12 in 0.006 seconds

	YEAR	MONTH	TOTAL
1	2016	1	7866
2	2016	2	10917
3	2016	3	13121
4	2016	4	50974
5	2016	5	16859
6	2016	6	18514
7	2016	7	20784
8	2016	8	38011
9	2016	9	31448
10	2016	10	9713
11	2016	11	10748
12	2016	12	14286

Query 5:

QUERY TO FIND THE TOP 10 CHEAPEST FLIGHTS FROM ONE LOCATION TO ANOTHER LOCATION ALONG WITH RELEVANT DETAILS.

```
SELECT FP.FLIGHTID,
       TO_CHAR(FP.PRICE,'$99,999.99') AS PRICE,
       FL.DEPARTDATE,
       TO_CHAR(FL.DEPARTTIME,'HH:MI:SS AM') AS DEPARTTIME,
       TO_CHAR(FL.ARRIVALTIME,'HH:MI:SS AM') AS ARRIVALTIME,
       AR1.CITY AS "DEPARTURE CITY",
       AR2.CITY AS "ARRIVAL CITY"
FROM FLIGHTPRICE FP
JOIN FLIGHT FL ON FL.FLIGHTID = FP.FLIGHTID
JOIN AIRPORT AR1 ON AR1.AIRPORTID = FL.DEPARTPORT
JOIN AIRPORT AR2 ON AR2.AIRPORTID = FL.ARRIVALPORT
ORDER BY FP.PRICE
OFFSET 20 ROWS FETCH NEXT 10 ROWS ONLY;
```

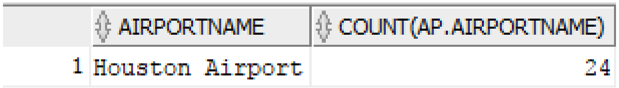


	FLIGHTID	PRICE	DEPARTDATE	DEPARTTIME	ARRIVALTIME	DEPARTURE CITY	ARRIVAL CITY
1	F-1916	$1,120.00	01-05-17	07:55:00 AM	06:16:00 AM	Katy	Bakersfield
2	F-2097	$1,171.00	25-10-17	05:00:00 PM	11:17:00 AM	Sioux City	San Jose
3	F-1930	$1,176.00	12-06-17	04:33:00 PM	02:17:00 AM	Orlando	Buffalo
4	F-1639	$1,231.00	31-07-17	03:01:00 AM	08:33:00 AM	Providence	Houston
5	F-1759	$1,305.00	03-10-17	06:10:00 AM	05:17:00 AM	Erie	Sacramento
6	F-1206	$1,443.00	01-11-17	10:25:00 PM	04:07:00 AM	Lansing	Chula Vista
7	F-1865	$1,467.00	24-08-17	09:53:00 PM	03:47:00 AM	Los Angeles	Gainesville
8	F-1933	$1,494.00	14-11-16	02:58:00 PM	02:16:00 PM	College Station	Buffalo
9	F-1413	$1,507.00	25-04-17	08:57:00 PM	07:15:00 PM	San Antonio	Stamford
10	F-1980	$1,511.00	01-06-17	08:35:00 AM	04:20:00 AM	Lexington	Atlanta

Query 6:

QUERY TO FIND THE BUSIEST AIRPORT WITH MOST NUMBER OF FLIGHTS DEPARTING

```
SELECT AP.AIRPORTNAME,
        COUNT(AP.AIRPORTNAME)
FROM FLIGHT F
INNER JOIN AIRPLANE A ON F.AIRPLANEID = A.AIRPLANEID
INNER JOIN AIRPORT AP ON A.AIRPORTID = AP.AIRPORTID
GROUP BY AP.AIRPORTNAME
HAVING COUNT(*) =
     (SELECT MAX(COUNT(*))
    FROM FLIGHT F
     INNER JOIN AIRPLANE A ON F.AIRPLANEID = A.AIRPLANEID
    INNER JOIN AIRPORT AP ON A.AIRPORTID = AP.AIRPORTID
    GROUP BY AP.AIRPORTNAME);
```



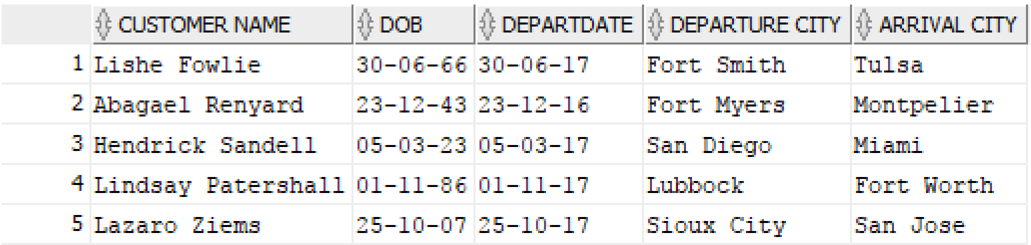
	AIRPORTNAME	COUNT(AP.AIRPORTNAME)
1	Houston Airport	24

Query 7:

QUERY TO FIND THE PASSENGER AND TRAVEL DETAILS FOR THOSE WHO ARE TRAVELLING ON THEIR BIRTHDAY

```
SELECT FNAME || ' ' || LNAME AS "CUSTOMER NAME",
       C.DOB,
       F.DEPARTDATE,
       AR1.CITY AS "DEPARTURE CITY",
       AR2.CITY AS "ARRIVAL CITY"
```

```
FROM CUSTOMER C
INNER JOIN RESERVATION R ON R.CUSTOMERID = C.CUSTOMERID
INNER JOIN FLIGHT F ON R.FLIGHTID = F.FLIGHTID
JOIN AIRPORT AR1 ON AR1.AIRPORTID = F.DEPARTPORT
JOIN AIRPORT AR2 ON AR2.AIRPORTID = F.ARRIVALPORT
WHERE TO_CHAR(C.DOB,'MMDD')=TO_CHAR(F.DEPARTDATE,'MMDD');
```



	CUSTOMER NAME	DOB	DEPARTDATE	DEPARTURE CITY	ARRIVAL CITY
1	Lishe Fowlie	30-06-66	30-06-17	Fort Smith	Tulsa
2	Abagael Renyard	23-12-43	23-12-16	Fort Myers	Montpelier
3	Hendrick Sandell	05-03-23	05-03-17	San Diego	Miami
4	Lindsay Patershall	01-11-86	01-11-17	Lubbock	Fort Worth
5	Lazaro Ziems	25-10-07	25-10-17	Sioux City	San Jose

Query 8:

QUERY TO FIND CUSTOMERS WHO MADE LAST MINUTE BOOKINGS i.e. ON THE SAME DAY OF TRAVEL.

```
SELECT C.CUSTOMERID,
       FNAME || ' ' || LNAME AS "CUSTOMER NAME",
       F.DEPARTDATE,
       I.INVOICEDATE
FROM CUSTOMER C
INNER JOIN RESERVATION R ON C.CUSTOMERID = R.CUSTOMERID
INNER JOIN INVOICE I ON R.CUSTOMERID = I.CUSTOMERID
INNER JOIN FLIGHT F ON R.FLIGHTID = F.FLIGHTID
WHERE F.DEPARTDATE = I.INVOICEDATE;
```

9. Database Programming

Procedure 1:

If a user wants to change his password, following procedure would set a new password for the user.

```
CREATE OR REPLACE PROCEDURE UPDATEPASSWORD (
       CUSTOMER_ID VARCHAR2,
       EMAIL_ID VARCHAR2,
       PASSWD VARCHAR2,
       NEWVAL OUT VARCHAR2
)
AS
BEGIN
  UPDATE ACCOUNT
  SET PSWD = PASSWD WHERE CUSTOMERID = CUSTOMER_ID AND EMAIL = EMAIL_ID;

  IF( SQL%ROWCOUNT >= 1 )
  THEN
    NEWVAL := 'Successfully Updated';
  ELSE
    NEWVAL := 'Enter Valid Net_Id and Customer_Id';
  END IF;
END UPDATEPASSWORD;
```

Procedure 2:

Following procedure would authenticate the username and password entered by the user when logging in.

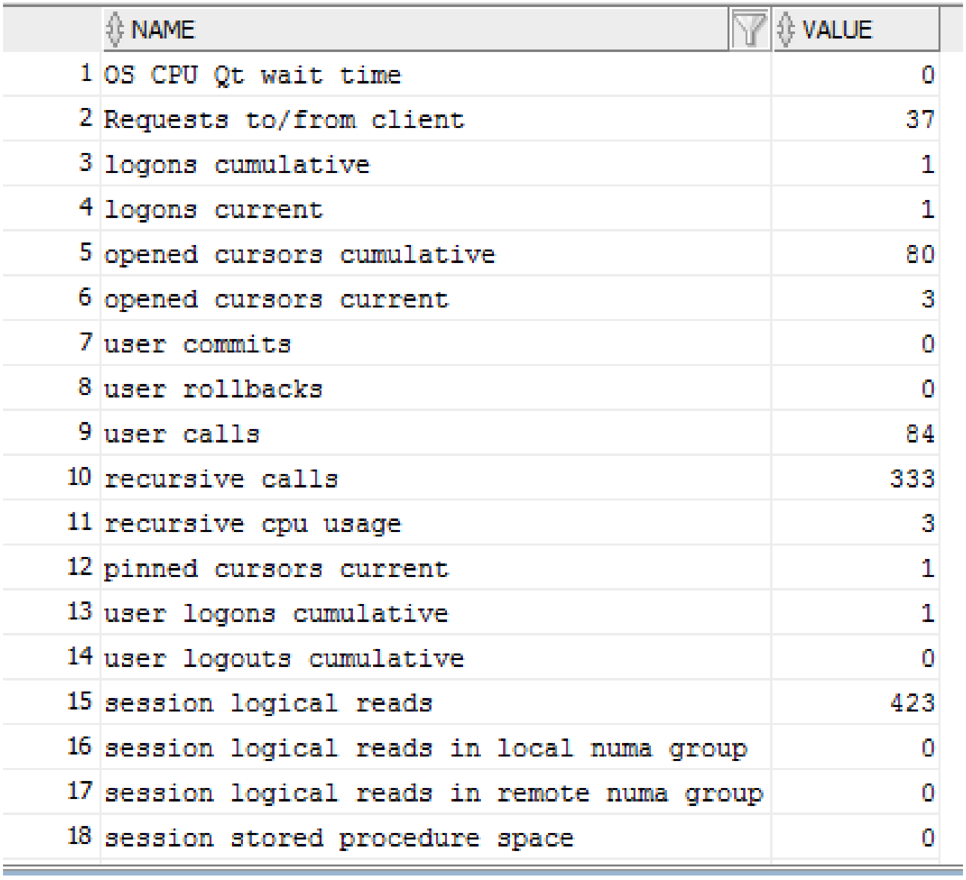
```
CREATE OR REPLACE FUNCTION log_in(username IN VARCHAR2, pw IN VARCHAR2)
RETURN VARCHAR2
AS
  count_match NUMBER;
BEGIN
  SELECT COUNT(*)
    INTO count_match
    FROM ACCOUNT
    WHERE customerid=username
    AND pswd=pw;
  IF count_match = 0 THEN
    dbms_output.put_line('Wrong username or password!');
  ELSIF count_match = 1 THEN
    dbms_output.put_line('Login successful!');
  ELSE
    dbms_output.put_line('Too many attempts!');
  END IF;
END;
```

10. DBA Scripts

Mostly Database administrators executes the DBA Scripts. Following are some of the scenarios where database scripts that are executed to control databases.

1. **To troubleshoot or identify any performance issue at any point of time, the administrator need to fetch the information about the session and its attributes. Following query will fetch this information.**

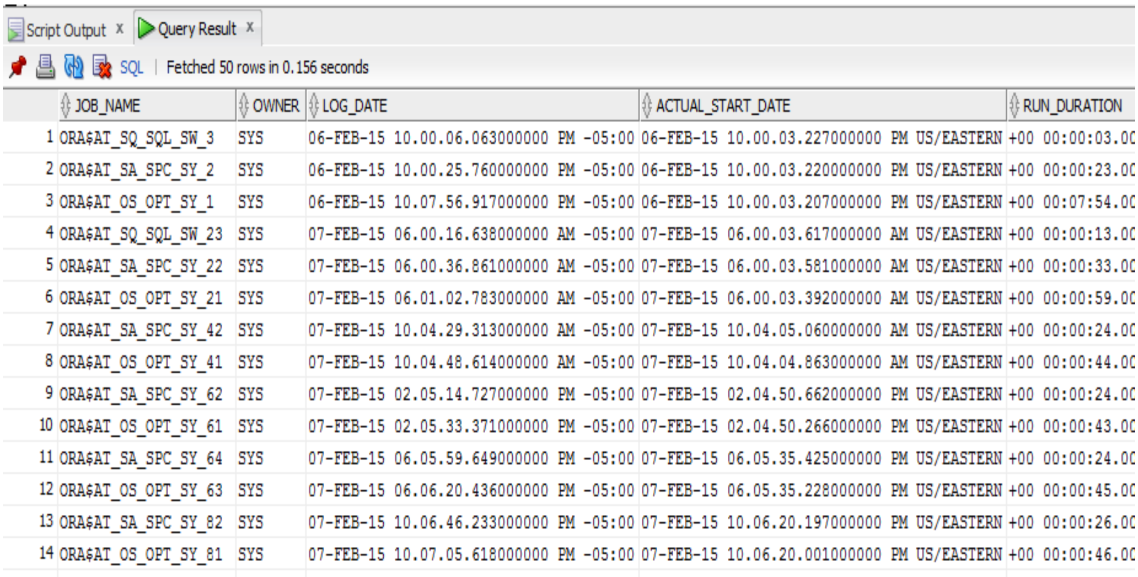
```
SELECT statsname.name, sstatistics.value
FROM
v$sesstat sstatistics,
v$statname statsname,
v$session sen WHERE
sstatistics.statistic# = statsname.statistic#
AND sen.audsid = SYS_CONTEXT ('USERENV','SESSIONID')
AND sen.sid = sstatistics.sid ;
```



	NAME	VALUE
1	OS CPU Qt wait time	0
2	Requests to/from client	37
3	logons cumulative	1
4	logons current	1
5	opened cursors cumulative	80
6	opened cursors current	3
7	user commits	0
8	user rollbacks	0
9	user calls	84
10	recursive calls	333
11	recursive cpu usage	3
12	pinned cursors current	1
13	user logons cumulative	1
14	user logouts cumulative	0
15	session logical reads	423
16	session logical reads in local numa group	0
17	session logical reads in remote numa group	0
18	session stored procedure space	0

2. **In order to identify database performance issues, session statistics must available to the DBA and following query will return all the session statistics.**

```
SELECT job_name,
owner,
log_date,
actual_start_date,
run_duration,
status
FROM
dba_scheduler_job_run_details ORDER BY log_date;
```

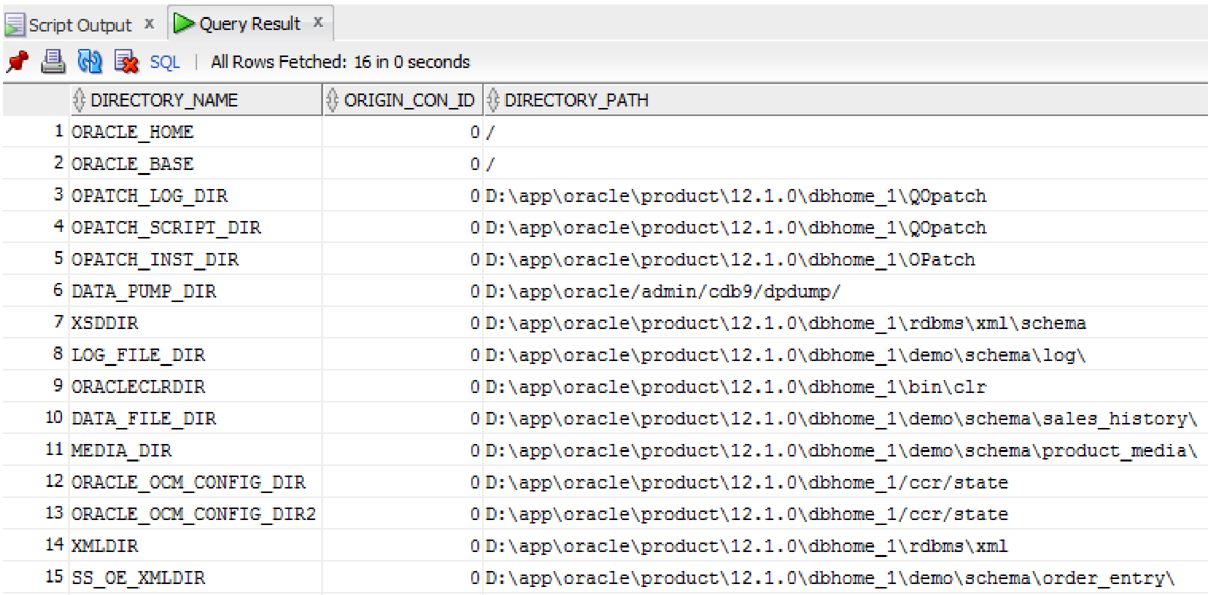


Fetched 50 rows in 0.156 seconds

	JOB_NAME	OWNER	LOG_DATE	ACTUAL_START_DATE	RUN_DURATION
1	ORA$AT_SQ_SQL_SW_3	SYS	06-FEB-15 10.00.06.063000000 PM -05:00	06-FEB-15 10.00.03.227000000 PM US/EASTERN	+00 00:00:03.0
2	ORA$AT_SA_SPC_SY_2	SYS	06-FEB-15 10.00.25.760000000 PM -05:00	06-FEB-15 10.00.03.220000000 PM US/EASTERN	+00 00:00:23.0
3	ORA$AT_OS_OPT_SY_1	SYS	06-FEB-15 10.07.56.917000000 PM -05:00	06-FEB-15 10.00.03.207000000 PM US/EASTERN	+00 00:07:54.0
4	ORA$AT_SQ_SQL_SW_23	SYS	07-FEB-15 06.00.16.638000000 AM -05:00	07-FEB-15 06.00.03.617000000 AM US/EASTERN	+00 00:00:13.0
5	ORA$AT_SA_SPC_SY_22	SYS	07-FEB-15 06.00.36.861000000 AM -05:00	07-FEB-15 06.00.03.581000000 AM US/EASTERN	+00 00:00:33.0
6	ORA$AT_OS_OPT_SY_21	SYS	07-FEB-15 06.01.02.783000000 AM -05:00	07-FEB-15 06.00.03.392000000 AM US/EASTERN	+00 00:00:59.0
7	ORA$AT_SA_SPC_SY_42	SYS	07-FEB-15 10.04.29.313000000 AM -05:00	07-FEB-15 10.04.05.060000000 AM US/EASTERN	+00 00:00:24.0
8	ORA$AT_OS_OPT_SY_41	SYS	07-FEB-15 10.04.48.614000000 AM -05:00	07-FEB-15 10.04.04.863000000 AM US/EASTERN	+00 00:00:44.0
9	ORA$AT_SA_SPC_SY_62	SYS	07-FEB-15 02.05.14.727000000 PM -05:00	07-FEB-15 02.04.50.662000000 PM US/EASTERN	+00 00:00:24.0
10	ORA$AT_OS_OPT_SY_61	SYS	07-FEB-15 02.05.33.371000000 PM -05:00	07-FEB-15 02.04.50.266000000 PM US/EASTERN	+00 00:00:43.0
11	ORA$AT_SA_SPC_SY_64	SYS	07-FEB-15 06.05.59.649000000 PM -05:00	07-FEB-15 06.05.35.425000000 PM US/EASTERN	+00 00:00:24.0
12	ORA$AT_OS_OPT_SY_63	SYS	07-FEB-15 06.06.20.436000000 PM -05:00	07-FEB-15 06.05.35.228000000 PM US/EASTERN	+00 00:00:45.0
13	ORA$AT_SA_SPC_SY_82	SYS	07-FEB-15 10.06.46.233000000 PM -05:00	07-FEB-15 10.06.20.197000000 PM US/EASTERN	+00 00:00:26.0
14	ORA$AT_OS_OPT_SY_81	SYS	07-FEB-15 10.07.05.618000000 PM -05:00	07-FEB-15 10.06.20.001000000 PM US/EASTERN	+00 00:00:46.0

3. **With following query, DBA can record the target path of the directories in the database, which can be helpful for the backups.**

```
SELECT directory_name,
ORIGIN_CON_ID,
directory_path
FROM
dba_directories;
```



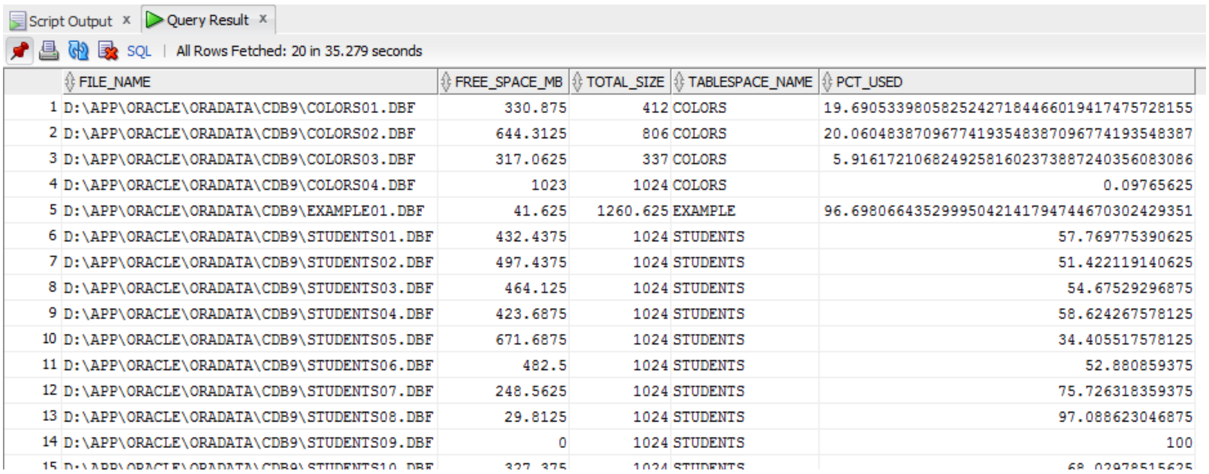
Script Output × | Query Result ×

SQL | All Rows Fetched: 16 in 0 seconds

	DIRECTORY_NAME	ORIGIN_CON_ID	DIRECTORY_PATH
1	ORACLE_HOME	0	/
2	ORACLE_BASE	0	/
3	OPATCH_LOG_DIR	0	D:\app\oracle\product\12.1.0\dbhome_1\QOpatch
4	OPATCH_SCRIPT_DIR	0	D:\app\oracle\product\12.1.0\dbhome_1\QOpatch
5	OPATCH_INST_DIR	0	D:\app\oracle\product\12.1.0\dbhome_1\OPatch
6	DATA_PUMP_DIR	0	D:\app\oracle/admin/cdb9/dpdump/
7	XSDDIR	0	D:\app\oracle\product\12.1.0\dbhome_1\rdbms\xml\schema
8	LOG_FILE_DIR	0	D:\app\oracle\product\12.1.0\dbhome_1\demo\schema\log\
9	ORACLECLRDIR	0	D:\app\oracle\product\12.1.0\dbhome_1\bin\clr
10	DATA_FILE_DIR	0	D:\app\oracle\product\12.1.0\dbhome_1\demo\schema\sales_history\
11	MEDIA_DIR	0	D:\app\oracle\product\12.1.0\dbhome_1\demo\schema\product_media\
12	ORACLE_OCM_CONFIG_DIR	0	D:\app\oracle\product\12.1.0\dbhome_1/ccr/state
13	ORACLE_OCM_CONFIG_DIR2	0	D:\app\oracle\product\12.1.0\dbhome_1/ccr/state
14	XMLDIR	0	D:\app\oracle\product\12.1.0\dbhome_1\rdbms\xml
15	SS_OE_XMLDIR	0	D:\app\oracle\product\12.1.0\dbhome_1\demo\schema\order_entry\

4. **Effective utilization of the memory is essential to improve performance of the database. A DBA can find out the free space available for current datafiles.**

```
Select df.file_name,
nvl(fr.bytes/(1024*1024),0) free_space_MB,
df.bytes/(1024*1024) total_size,
df.tablespace_name,
((df.bytes-nvl(fr.bytes,0))/df.bytes)*100 pct_used
from (select sum(bytes) bytes, file_id from dba_free_space group by
file_id) fr, dba_data_files df
where df.file_id = fr.file_id(+) order by 1, df.file_id;
```



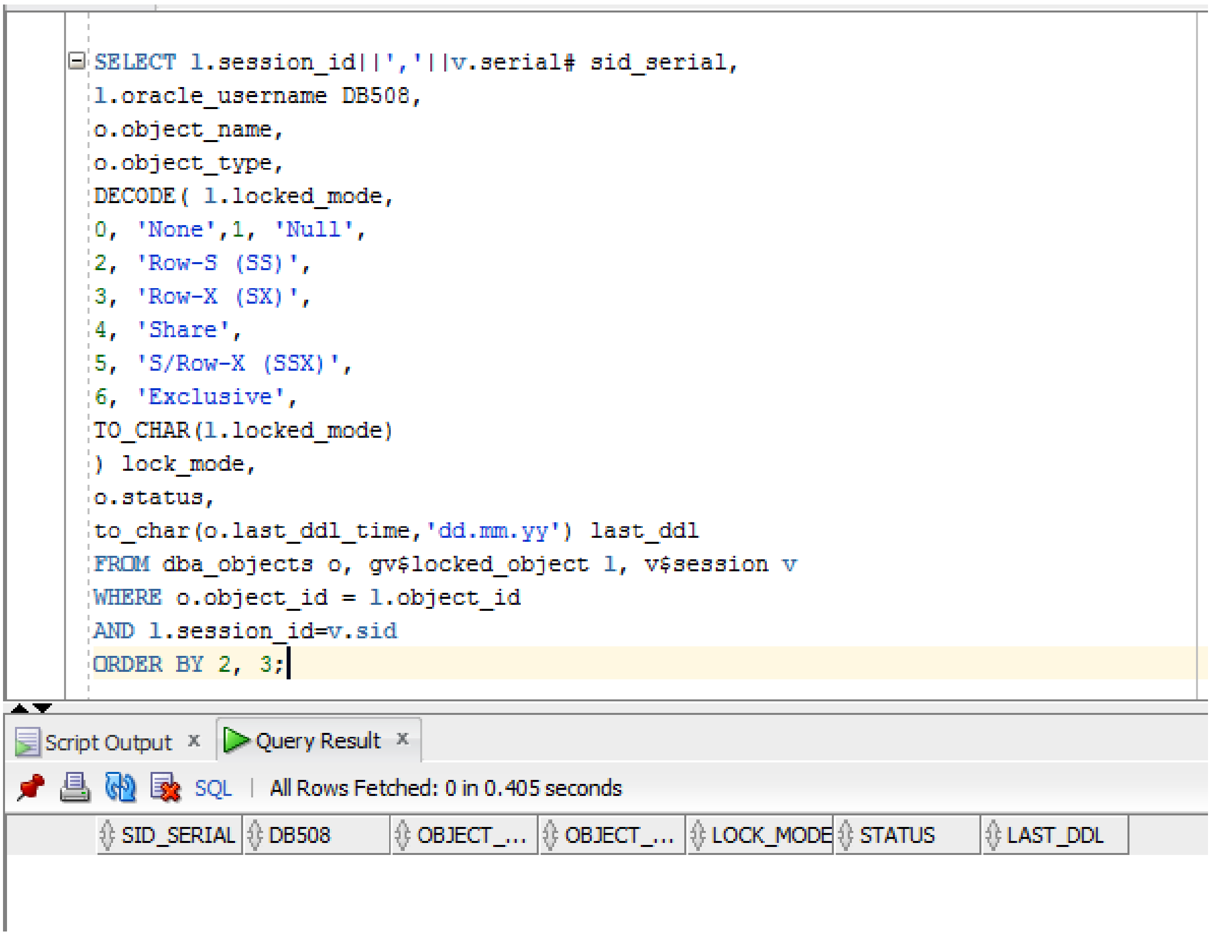
Script Output × | Query Result ×

SQL | All Rows Fetched: 20 in 35.279 seconds

	FILE_NAME	FREE_SPACE_MB	TOTAL_SIZE	TABLESPACE_NAME	PCT_USED
1	D:\APP\ORACLE\ORADATA\CDB9\COLORS01.DBF	330.875	412	COLORS	19.69053398058252427184466019417475728155
2	D:\APP\ORACLE\ORADATA\CDB9\COLORS02.DBF	644.3125	806	COLORS	20.06048387096774193548387096774193548387
3	D:\APP\ORACLE\ORADATA\CDB9\COLORS03.DBF	317.0625	337	COLORS	5.91617210682492581602373887240356083086
4	D:\APP\ORACLE\ORADATA\CDB9\COLORS04.DBF	1023	1024	COLORS	0.09765625
5	D:\APP\ORACLE\ORADATA\CDB9\EXAMPLE01.DBF	41.625	1260.625	EXAMPLE	96.69806643529995042141794744670302429351
6	D:\APP\ORACLE\ORADATA\CDB9\STUDENTS01.DBF	432.4375	1024	STUDENTS	57.769775390625
7	D:\APP\ORACLE\ORADATA\CDB9\STUDENTS02.DBF	497.4375	1024	STUDENTS	51.422119140625
8	D:\APP\ORACLE\ORADATA\CDB9\STUDENTS03.DBF	464.125	1024	STUDENTS	54.67529296875
9	D:\APP\ORACLE\ORADATA\CDB9\STUDENTS04.DBF	423.6875	1024	STUDENTS	58.624267578125
10	D:\APP\ORACLE\ORADATA\CDB9\STUDENTS05.DBF	671.6875	1024	STUDENTS	34.405517578125
11	D:\APP\ORACLE\ORADATA\CDB9\STUDENTS06.DBF	482.5	1024	STUDENTS	52.880859375
12	D:\APP\ORACLE\ORADATA\CDB9\STUDENTS07.DBF	248.5625	1024	STUDENTS	75.726318359375
13	D:\APP\ORACLE\ORADATA\CDB9\STUDENTS08.DBF	29.8125	1024	STUDENTS	97.088623046875
14	D:\APP\ORACLE\ORADATA\CDB9\STUDENTS09.DBF	0	1024	STUDENTS	100
15	D:\APP\ORACLE\ORADATA\CDB9\STUDENTS10.DBF	327.375	1024	STUDENTS	68.02978515625

5. **DBA can troubleshoot by identifying locked sessions with following query, it will display the name of the user that created the lock, object on which the lock is created, session id and serial number.**

```
SELECT l.session_id||','||v.serial# sid_serial,
l.oracle_username DB508,
o.object_name,
o.object_type,
DECODE( l.locked_mode,
0, 'None',1, 'Null',
2, 'Row-S (SS)',
3, 'Row-X (SX)',
4, 'Share',
5, 'S/Row-X (SSX)',
6, 'Exclusive',
TO_CHAR(l.locked_mode)
) lock_mode,
o.status,
to_char(o.last_ddl_time,'dd.mm.yy') last_ddl
FROM dba_objects o, gv$locked_object l, v$session v
WHERE o.object_id = l.object_id
AND l.session_id=v.sid
ORDER BY 2, 3;
```



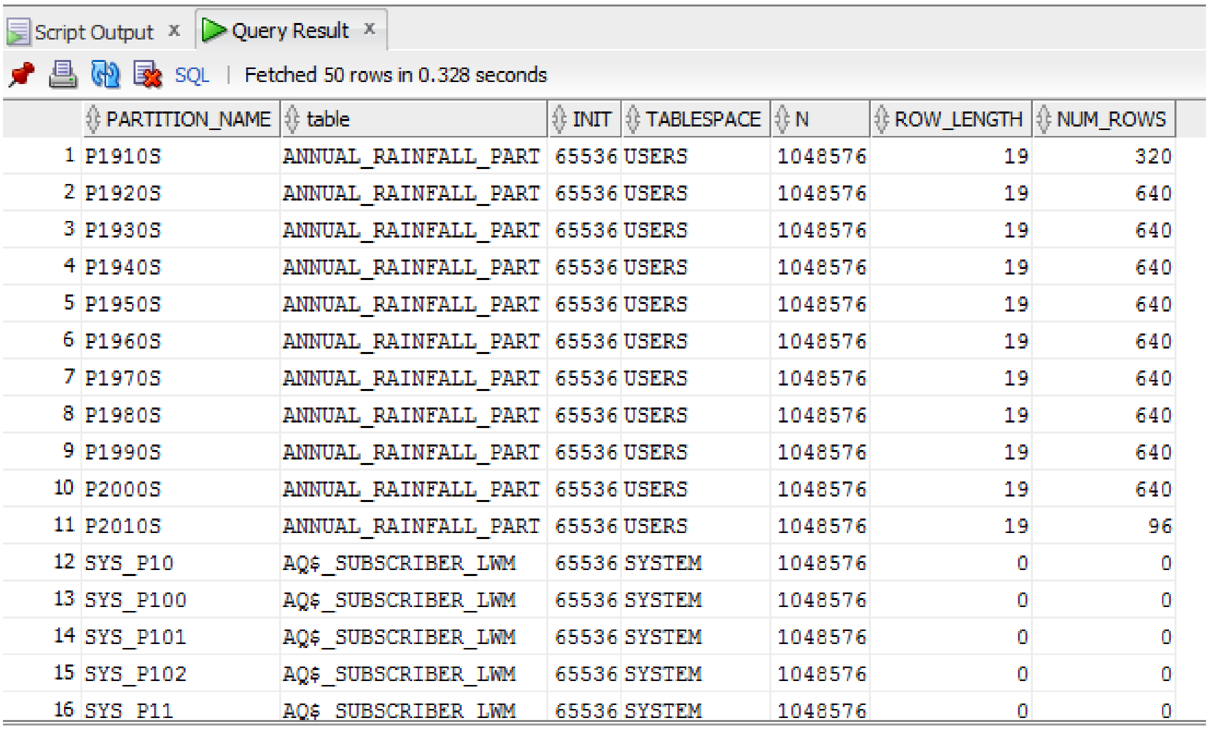
```
SELECT l.session_id||','||v.serial# sid_serial,
l.oracle_username DB508,
o.object_name,
o.object_type,
DECODE( l.locked_mode,
0, 'None',1, 'Null',
2, 'Row-S (SS)',
3, 'Row-X (SX)',
4, 'Share',
5, 'S/Row-X (SSX)',
6, 'Exclusive',
TO_CHAR(l.locked_mode)
) lock_mode,
o.status,
to_char(o.last_ddl_time,'dd.mm.yy') last_ddl
FROM dba_objects o, gv$locked_object l, v$session v
WHERE o.object_id = l.object_id
AND l.session_id=v.sid
ORDER BY 2, 3;
```

All Rows Fetched: 0 in 0.405 seconds

SID_SERIAL	DB508	OBJECT_...	OBJECT_...	LOCK_MODE	STATUS	LAST_DDL

6. **For effective space management, with following query DBA can know the partition details of the table sorted in order of corresponding row length.**

```
SELECT part.partition_name,
part.table_name as "table",
part.initial_extent init,
part.tablespace_name tablespace,
part.next_extent n,
part.avg_row_len row_length,
part.num_rows
FROM dba_tab_partitions part
ORDER BY part.table_name, part.partition_name;
```



Fetched 50 rows in 0.328 seconds

	PARTITION_NAME	table	INIT	TABLESPACE	N	ROW_LENGTH	NUM_ROWS
1	P1910S	ANNUAL_RAINFALL_PART	65536	USERS	1048576	19	320
2	P1920S	ANNUAL_RAINFALL_PART	65536	USERS	1048576	19	640
3	P1930S	ANNUAL_RAINFALL_PART	65536	USERS	1048576	19	640
4	P1940S	ANNUAL_RAINFALL_PART	65536	USERS	1048576	19	640
5	P1950S	ANNUAL_RAINFALL_PART	65536	USERS	1048576	19	640
6	P1960S	ANNUAL_RAINFALL_PART	65536	USERS	1048576	19	640
7	P1970S	ANNUAL_RAINFALL_PART	65536	USERS	1048576	19	640
8	P1980S	ANNUAL_RAINFALL_PART	65536	USERS	1048576	19	640
9	P1990S	ANNUAL_RAINFALL_PART	65536	USERS	1048576	19	640
10	P2000S	ANNUAL_RAINFALL_PART	65536	USERS	1048576	19	640
11	P2010S	ANNUAL_RAINFALL_PART	65536	USERS	1048576	19	96
12	SYS_P10	AQ$_SUBSCRIBER_LWM	65536	SYSTEM	1048576	0	0
13	SYS_P100	AQ$_SUBSCRIBER_LWM	65536	SYSTEM	1048576	0	0
14	SYS_P101	AQ$_SUBSCRIBER_LWM	65536	SYSTEM	1048576	0	0
15	SYS_P102	AQ$_SUBSCRIBER_LWM	65536	SYSTEM	1048576	0	0
16	SYS_P11	AQ$_SUBSCRIBER_LWM	65536	SYSTEM	1048576	0	0

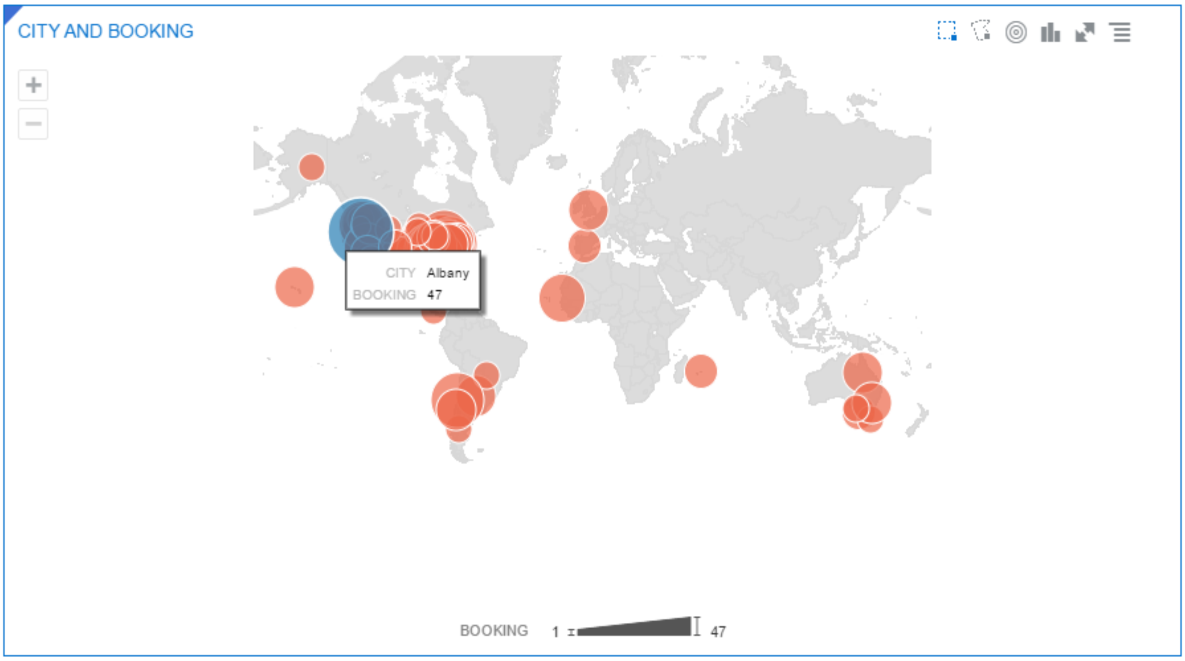
11. Data visualization

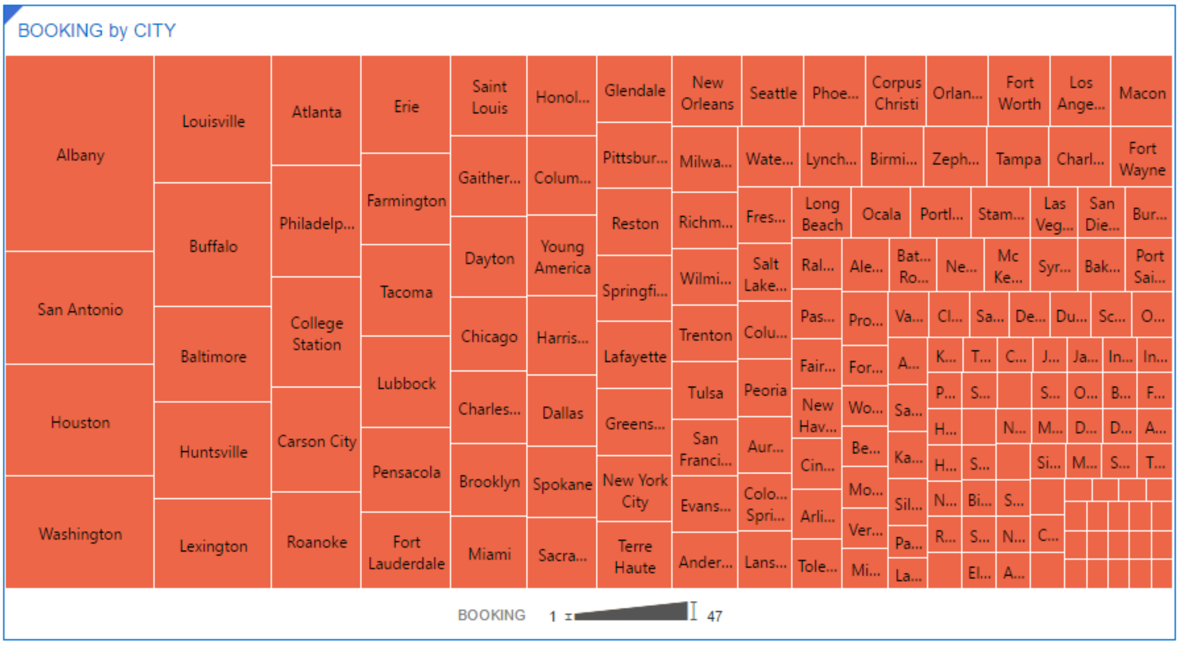
Data visualization have been done suing Oracle Data Visualization Desktop. We first connected the airline database to the application and created following visualizations.

1. **Busiest City**

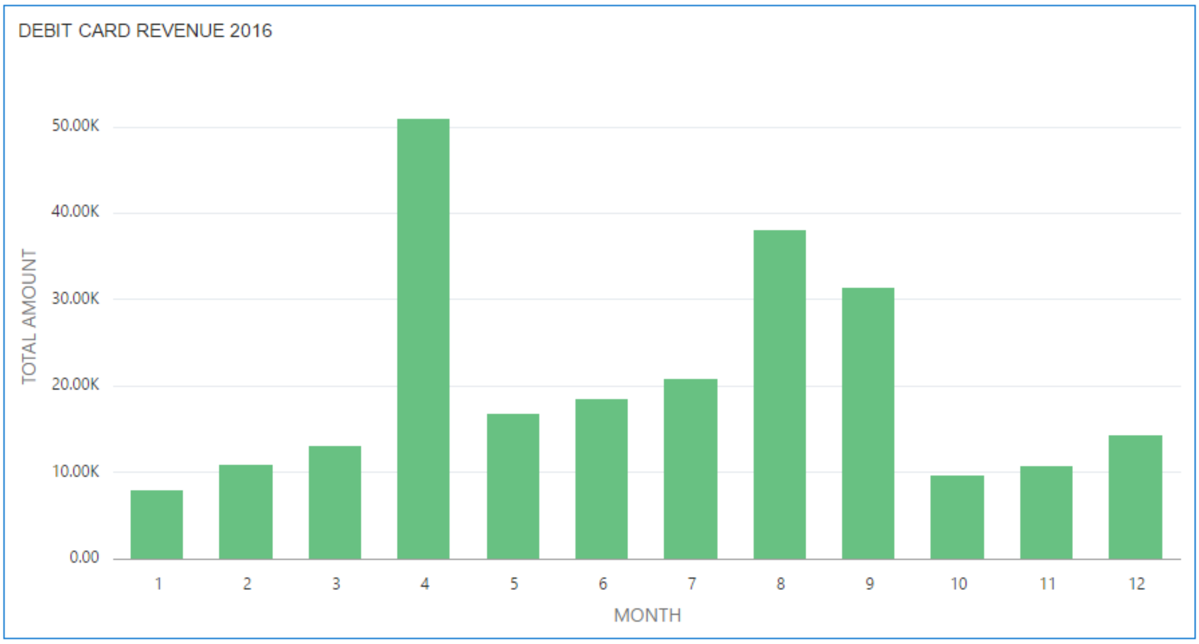
   The city from which most of the people fly from.

   In the below graph Albany have been the busiest port with 47 reservations.





2. Monthly amount incurred using debit card in 2016.



3. Monthly amount incurred using credit card in 2016

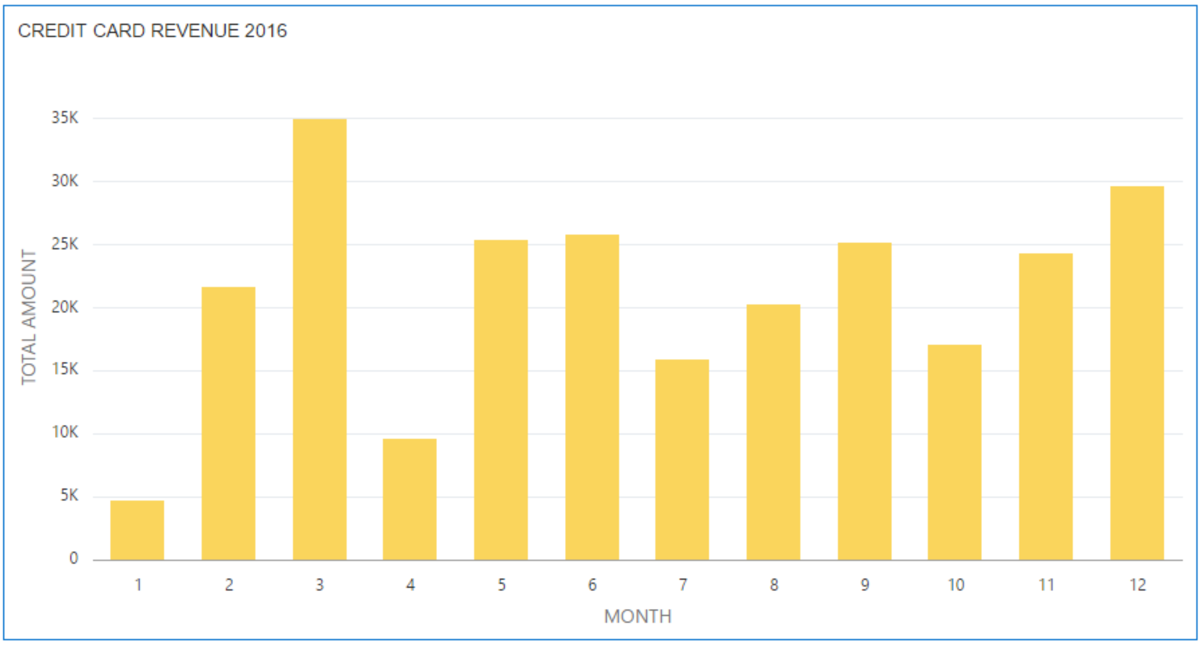


Table of Evaluation

- Logical database design **– 25 Points**
- Physical database design - **17 Points**
- Data generation and loading – **20 Points**
- Performance tuning – **8 Points**
- Querying – **12 Points**
- DBA scripts – **9 Points**
- Data visualization – **4 Points**
- Database programming (Stored Procedure) – **5 Points**