**EXECUTIVE SUMMARY**

**Problem Statement**

Cardinal Health is a global, integrated healthcare services and products company, providing customized solutions for hospitals, health systems, pharmacies. The company has many dealerships with pharmaceutical companies from which the medications are purchased directly. Then the company will supply the medications to local pharmacies per order request. For the past few months they are facing difficulties in the current system for tracking down the details of the medication transactions due to increased consumption of the medicines. Therefore, they are looking for a solution to resolve difficulties in keeping tracking and maintaining the records of medication transactions. They are also looking for an efficient way to manage employees.

**The main purposes of proposed system :**

* Management of buying and selling medications.
* Management of Stock.
* Management of employees.

**System’s Objects:**

* Employees/Administrator
* Customers

**Solution**

The solution is the creation of a desktop application and multiple database tables include the relationships between those tables. The tables consist of the following: **Medication**, **Customer**, **Customer\_Order**, **Manufacturer**, **Manufacturer\_Order**, and **Employee**. These tables represent the main entities of the application and define relationships between entities. We can create queries and reports that will provide the necessary information. The database will be implemented using Oracle 11g.

**Benefits to Users**

* Users will be able to order bulk medication from manufacturers.
* Users will be able to manage orders from customers such as CVS Pharmacy, Kroger, Walgreens, etc.
* Users will be able to manage the company’s employees and have access to their information.
* Finding and management of information about medication, manufacturers, customers, employees, and orders correctly and quickly. For customers living in another locations, information will be updated regularly through the website, so customers can find information of medication suitable and fast.

**Hardware/Software Requirements:**

* Hardware**:**
* Intel i5 processor
* 4 GB of RAM or greater
* Windows 7 or later
* Java Virtual Machine
* Software: Either or combination of the following
* JDK 8 or later/.NET/J2EE
* IDE
* Oracle SQL Developer/MySQL/Microsoft SQL Server

**Project Outline**

The project will contain the following major components:

* Schema design
* Entity-Relationship Diagram
* Table implementation
* Queries
* Reports

**Part II. Schema Design**

**Medication** (NDC, Medication\_Name, Manu\_ID, Made\_In\_Country, Unit)

FK Manu\_ID 🡪 Manufacturer

**Customer** (Customer\_ID,Last\_Name,First\_Name,Street,City,State,Zip,Fax,Phone\_Number,Tax\_ID,Web\_Site,Rate)

**Order** (Customer\_ID,NDC,Quantity,Total\_Price,Order\_Date,Delivery\_Date,Expiration\_Date,Paid,Emp\_Ordered)

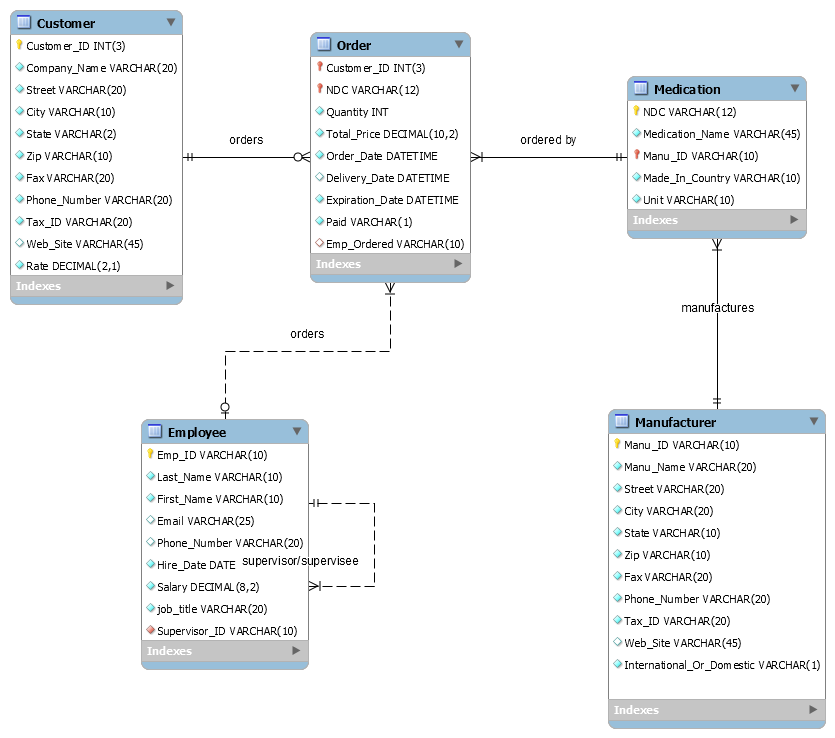
FK Customer\_ID 🡪 Customer  
 FK NDC 🡪 Medication  
 FK Emp\_Ordered 🡪 Employee

**Manufacturer** (Manu\_ID,Manu\_Name,Street,City,State,Zip,Fax,Phone\_Number,Email,Tax\_ID,Web\_Site, International\_or\_Domestic)

**Employee** (Emp\_ID,Last\_Name,First\_Name,Email,Phone\_Number,Hire\_Date,Salary,Role,Supervisor\_ID)

FK Supervisor\_ID 🡪 Employee

**Entity-Relationship Diagram**

****

--Dropping TABLES for multiple runs

DROP TABLE cardinal\_orders CASCADE CONSTRAINTS;

DROP TABLE cardinal\_medication CASCADE CONSTRAINTS;

DROP TABLE cardinal\_manufacturers CASCADE CONSTRAINTS;

DROP TABLE cardinal\_employees CASCADE CONSTRAINTS;

DROP TABLE cardinal\_customers CASCADE CONSTRAINTS;

DROP SEQUENCE cardinal\_customers\_seq;

--Creating CARDINAL\_CUSTOMERS table

CREATE TABLE cardinal\_customers

( customer\_id number(3),

company\_name varchar2(20) NOT NULL,

street varchar2(20) NOT NULL,

city varchar2(10) NOT NULL,

state varchar2(2) NOT NULL,

zip varchar2(10) NOT NULL,

fax varchar2(20) NOT NULL,

phone\_number varchar2(20) NOT NULL,

tax\_id varchar2(20) NOT NULL,

web\_site varchar2(45),

rate number(2,1) NOT NULL,

CONSTRAINT pk\_customer\_id PRIMARY KEY (customer\_id)

);

--Creating the Primary Key sequence for CARDINAL\_CUSTOMERS

CREATE SEQUENCE cardinal\_customers\_seq

START WITH 1

INCREMENT BY 1;

--Creating CARDINAL\_MANUFACTURERS table

CREATE TABLE cardinal\_manufacturers

( manu\_id varchar2(10),

manu\_name varchar2(20) NOT NULL,

street varchar2(20) NOT NULL,

city varchar2(20) NOT NULL,

state varchar2(10) NOT NULL,

zip varchar2(10) NOT NULL,

fax varchar2(20) NOT NULL,

phone\_number varchar2(20) NOT NULL,

tax\_id varchar2(20) NOT NULL,

web\_site varchar2(30),

international\_or\_domestic varchar2(1) NOT NULL,

CONSTRAINT pk\_manu\_id PRIMARY KEY (manu\_id),

CONSTRAINT int\_dom\_check CHECK (international\_or\_domestic IN ('I','D'))

);

--Creating CARDINAL\_EMPLOYEES table

CREATE TABLE cardinal\_employees

( emp\_id varchar2(10),

first\_name varchar2(10) NOT NULL,

last\_name varchar2(10) NOT NULL,

email varchar2(25),

phone\_number varchar2(20),

hire\_date DATE NOT NULL,

salary number(8,2) NOT NULL,

job\_title varchar2(20) NOT NULL,

supervisor\_id varchar2(10),

CONSTRAINT pk\_employee\_id PRIMARY KEY (emp\_id),

CONSTRAINT fk\_supervisor\_id

FOREIGN KEY (emp\_id)

REFERENCES cardinal\_employees(emp\_id)

);

--Creating CARDINAL\_MEDICATION table

CREATE TABLE cardinal\_medication

( ndc varchar2(12),

medication\_name varchar2(45) NOT NULL,

manu\_id varchar2(10),

made\_in\_country varchar2(10) NOT NULL,

unit varchar2(10) NOT NULL,

CONSTRAINT pk\_medication PRIMARY KEY (ndc),

CONSTRAINT fk\_medication\_manu

FOREIGN KEY (manu\_id)

REFERENCES cardinal\_manufacturers(manu\_id)

);

--Creating CARDINAL\_ORDERS table

CREATE TABLE cardinal\_orders

( customer\_id number(3),

ndc varchar2(12),

quantity number NOT NULL,

total\_price number(10,2) NOT NULL,

order\_date DATE NOT NULL,

delivery\_date DATE,

expiration\_date DATE NOT NULL,

paid varchar2(1) NOT NULL,

emp\_ordered varchar2(10),

CONSTRAINT pk\_order PRIMARY KEY (customer\_id,ndc),

CONSTRAINT fk\_order\_customers

FOREIGN KEY (customer\_id)

REFERENCES cardinal\_customers(customer\_id),

CONSTRAINT fk\_order\_medication

FOREIGN KEY (ndc)

REFERENCES cardinal\_medication(ndc),

CONSTRAINT fk\_order\_employees

FOREIGN KEY (emp\_ordered)

REFERENCES cardinal\_employees(emp\_id),

CONSTRAINT paid\_check CHECK (paid IN ('y','n','Y','N')),

CONSTRAINT delivery\_date\_check CHECK (delivery\_date > order\_date),

CONSTRAINT expiration\_date\_check CHECK (expiration\_date > order\_date)

);

COMMIT;

**RUN OF SCRIPT:**

Table CARDINAL\_ORDERS dropped.

Table CARDINAL\_MEDICATION dropped.

Table CARDINAL\_MANUFACTURERS dropped.

Table CARDINAL\_EMPLOYEES dropped.

Table CARDINAL\_CUSTOMERS dropped.

Sequence cardinal\_customers\_seq dropped.

Table CARDINAL\_CUSTOMERS created.

Sequence cardinal\_customers\_seq created.

Table CARDINAL\_MANUFACTURERS created.

Table CARDINAL\_EMPLOYEES created.

Table CARDINAL\_MEDICATION created.

Table CARDINAL\_ORDERS created.

Commit complete.

--Part 4 Q1: UNION with JOIN query

--This query returns medication's NDC, medication's name, and manufacturer's name

--that have been ordered by customers.

SELECT m.ndc AS "NDC",

m.medication\_name AS "Medication",

ma.manu\_name AS "Manufacturer"

FROM cardinal\_medication m, cardinal\_manufacturers ma

WHERE m.manu\_id = ma.manu\_id

UNION

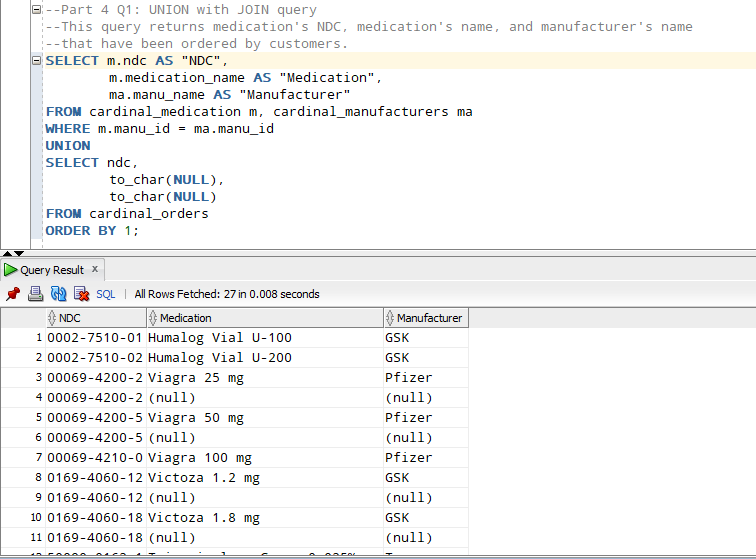
SELECT ndc,

to\_char(NULL),

to\_char(NULL)

FROM cardinal\_orders

ORDER BY 1;



--Part 4 Q2: INTERSECT

--This query returns the list of IDs of customers who have ordered medication.

SELECT customer\_id AS "Customer ID"

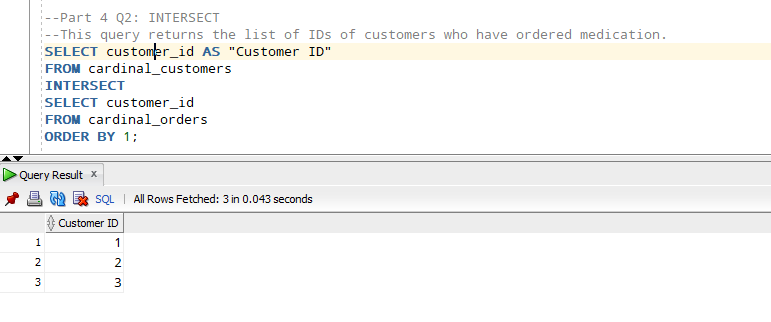
FROM cardinal\_customers

INTERSECT

SELECT customer\_id

FROM cardinal\_orders

ORDER BY 1;



--Part 4 Q3: MINUS

--This query returns the manufacturer that does not have any medication available to purchase through Cardinal.

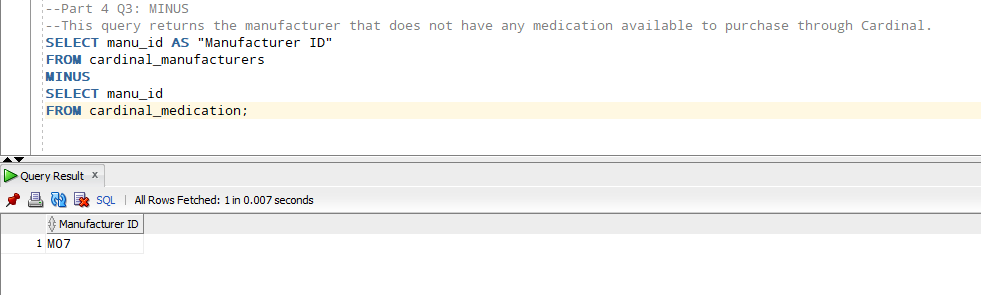
SELECT manu\_id AS "Manufacturer ID"

FROM cardinal\_manufacturers

MINUS

SELECT manu\_id

FROM cardinal\_medication;



--Query 1:

/\*This query returns employee id, first name, last name, and hire date

for all employees hired before 01/01/2016 and have salary less than 6000. Sort in the order of employee ID\*/

SELECT emp\_id as "Employee ID",

first\_name as "First Name",

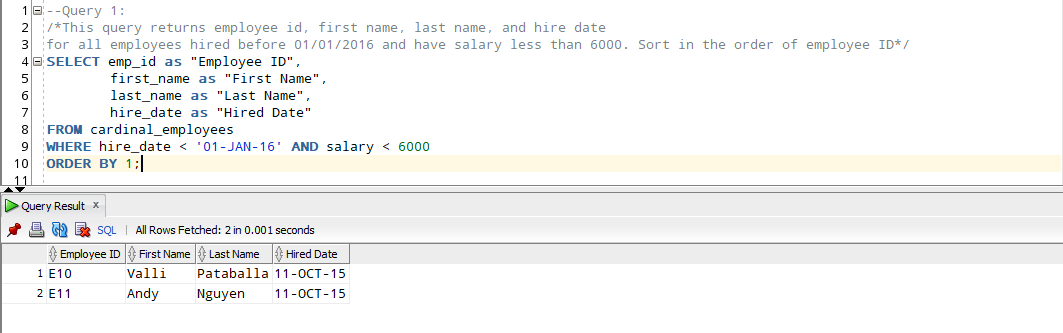
last\_name as "Last Name",

hire\_date as "Hired Date"

FROM cardinal\_employees

WHERE hire\_date < '01-JAN-16' AND salary < 6000

ORDER BY 1;



--Query 2:

/\*This query returns manufacturer name, address, and check if the manufacturer is domestic or international.

Sort in the order of manufacturer name\*/

SELECT manu\_name AS "Manufacturer Name",

street || ', ' || city || ', ' || state || ' ' || zip AS "Address",

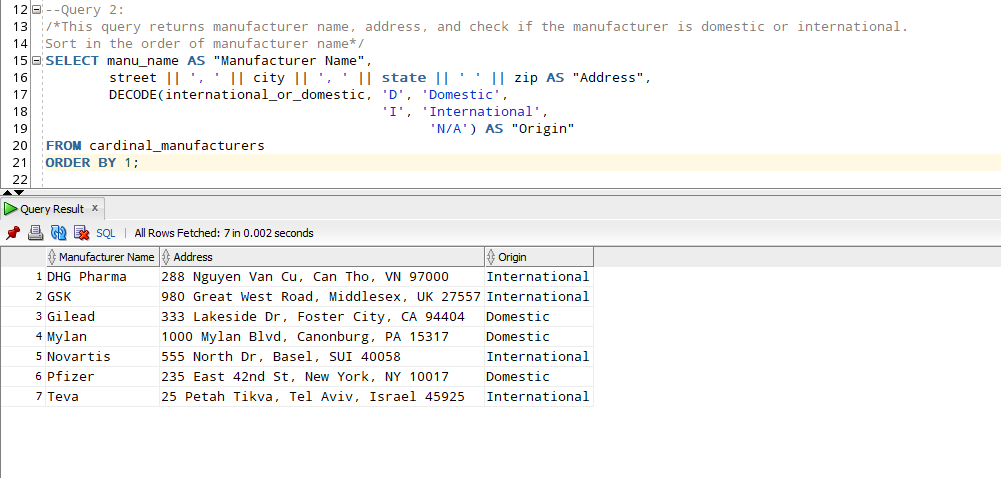
DECODE(international\_or\_domestic, 'D', 'Domestic',

'I', 'International',

'N/A') AS "Origin"

FROM cardinal\_manufacturers

ORDER BY 1;



--Query 3:

/\*This query returns the average order amount for each customer.

Sort in the order of order amount.\*/

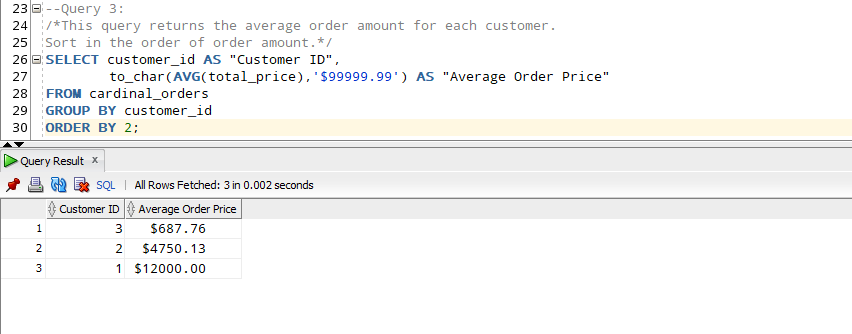
SELECT customer\_id AS "Customer ID",

to\_char(AVG(total\_price),'$99999.99') AS "Average Order Price"

FROM cardinal\_orders

GROUP BY customer\_id

ORDER BY 2;



--Query 4:

/\*This query returns ndc, medication name, manufacturer name, and unit of all medication in database.

Sort in order of medication name\*/

SELECT m.ndc AS "NDC",

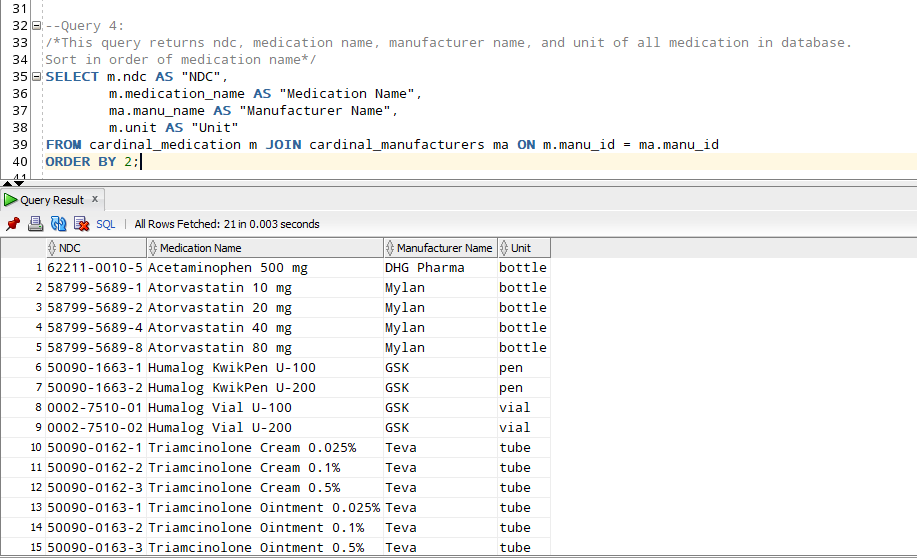
m.medication\_name AS "Medication Name",

ma.manu\_name AS "Manufacturer Name",

m.unit AS "Unit"

FROM cardinal\_medication m JOIN cardinal\_manufacturers ma ON m.manu\_id = ma.manu\_id

ORDER BY 2;



--Query 5:

/\*This query returns employee id, first name, last name, and salary of all employees that have a salary greater than

the average salary of the company. Sort in order of salary\*/

SELECT emp\_id AS "Employee ID",

first\_name AS "First Name",

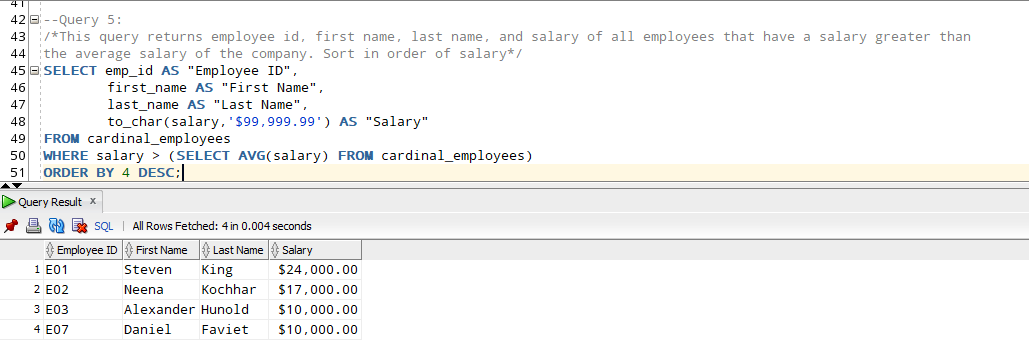
last\_name AS "Last Name",

to\_char(salary,'$99,999.99') AS "Salary"

FROM cardinal\_employees

WHERE salary > (SELECT AVG(salary) FROM cardinal\_employees)

ORDER BY 4 DESC;



--Query 6:

/\*This query returns employee id, first name, last name, job title, and salary

of the lowest paid employee for that supervisor. Exclude whose supervisor is not known.

Exclude groups where salary is less than 6000. Sort in order of salary.\*/

SELECT emp\_id AS "Supervisor ID",

first\_name || ' ' || last\_name AS "Name",

job\_title AS "Job Title",

MIN(salary) AS "Smallest Salary in Group"

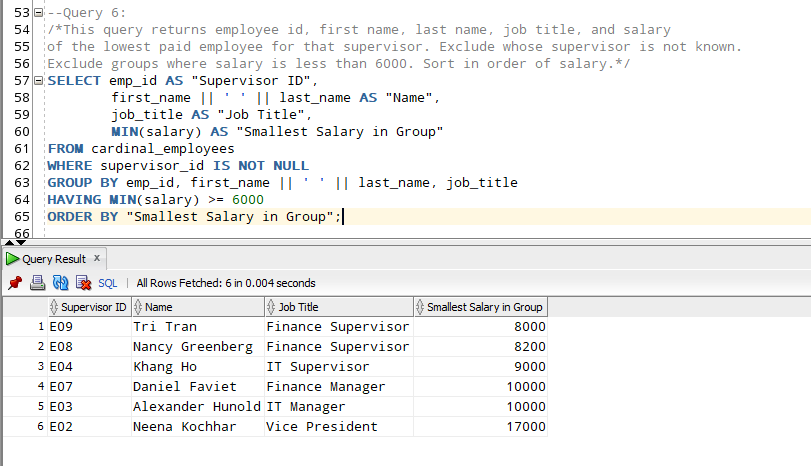
FROM cardinal\_employees

WHERE supervisor\_id IS NOT NULL

GROUP BY emp\_id, first\_name || ' ' || last\_name, job\_title

HAVING MIN(salary) >= 6000

ORDER BY "Smallest Salary in Group";



--Query 7:

/\*This query returns customer id, NDC, quantity, oder date, and calculates the expected delivery date.

The delivery date would be the first Monday following the order date. Sort in the ascending order of order date\*/

SELECT customer\_id AS "Customer ID",

ndc AS "NDC",

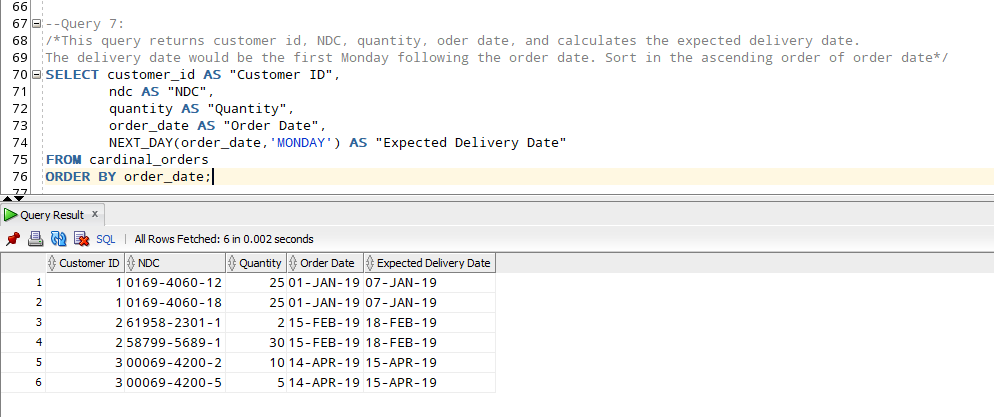
quantity AS "Quantity",

order\_date AS "Order Date",

NEXT\_DAY(order\_date,'MONDAY') AS "Expected Delivery Date"

FROM cardinal\_orders

ORDER BY order\_date;



--Query 8:

/\*This query returns first name, last name, and hire date of all employees.

Format the hire date in Month, day, yyyy (ex: January 1st, 2019). Sort in order of hire date\*/

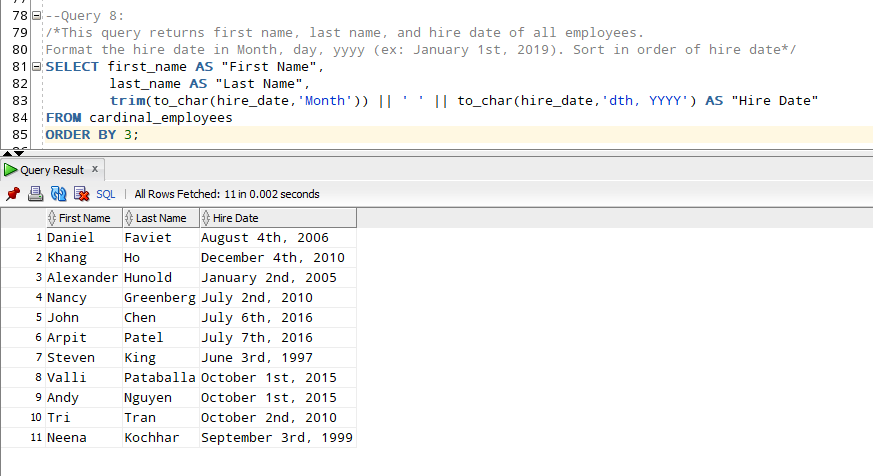
SELECT first\_name AS "First Name",

last\_name AS "Last Name",

trim(to\_char(hire\_date,'Month')) || ' ' || to\_char(hire\_date,'dth, YYYY') AS "Hire Date"

FROM cardinal\_employees

ORDER BY 3;



--Query 9:

/\*This query returns employee id, first name, last name, and salary

of all employees who have salary equal to average salary of their department (job title).

Sort in order of salary\*/

SELECT emp\_id AS "Employee ID",

first\_name AS "First Name",

last\_name AS "Last Name",

to\_char(salary,'$99,999') AS "Salary"

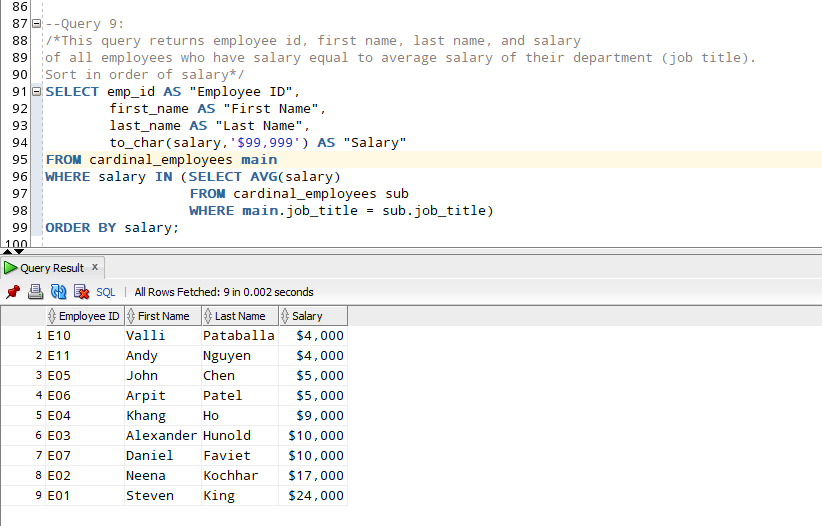
FROM cardinal\_employees main

WHERE salary IN (SELECT AVG(salary)

FROM cardinal\_employees sub

WHERE main.job\_title = sub.job\_title)

ORDER BY salary;



--Query 10:

/\*This query returns customer ID and customer name of all customers who have not ordered medication from Cardinal.

Sort in the order of customer ID.\*/

SELECT c.customer\_id AS "Customer ID",

c.company\_name AS "Customer Name"

FROM cardinal\_customers c LEFT OUTER JOIN cardinal\_orders o

ON c.customer\_id = o.customer\_id WHERE o.customer\_id IS NULL

ORDER BY 1;

