**PROJECT NAME**: WIRELESS NETWORK SERVICES MANAGEMENT SYSTEM

**TEAM NAME**: TEAM ACES

**PROJECT DEFINITION:**

The United States has a variety of wireless service providers. They provide different data plans, smartphones, mobile phones, internet gateway, hotspot devices, wristwatches, headphones, and accessories for mobile phones such as screen protectors, cases, cables, adaptors, etc. Every wireless service provider has a network of authorized dealer stores, each of which has a number of employees and a wide variety of devices. Only if the customers buy their products from wireless network service providers and subscribe to their services or switch lines from another service provider to their service then they will be able to operate their businesses. The management of all these tasks without information loss, providing the greatest customer service, and maintaining a profitable business is quite challenging for service providers. It is only possible if they maintain the data in a more systematic way. This inventory management system is useful both for the customer and dealers as it helps to retrieve any particular information by simply running the backend queries, it reduces the time taken by various transactions as they are effectively stored at the right place. By using the right database, multiple changes to information like customer details, customer purchase history, feedback details, advanced information regarding wireless services can be easily updated.

As part of this project, we will work on inventory management system for wireless network service providers. We will delve further into understanding the connections between various tables and properly format them. In this project, we can create many tables, such as the customer table, which contains the customer's first and last name, address, city, state, zip code, email address, and phone number. Dealer table containing information about the name, address, quantity of devices shipped, and personnel. Employee table containing the following information: employee id, first and last names, address, city, state, zip code, email address, and phone number. The invoice table contains the item id, sold price, payment method, employee name who sold the item, transaction reference number, and so on. Payroll table with employee id, date of hire, number of days worked in a week, hours worked, days off in a week, and so forth. Feedback table for employers which contains information such as employee id, star ratings, feedback comments on the service provided. We will add more tables as we go further into the details and devote more time to it.
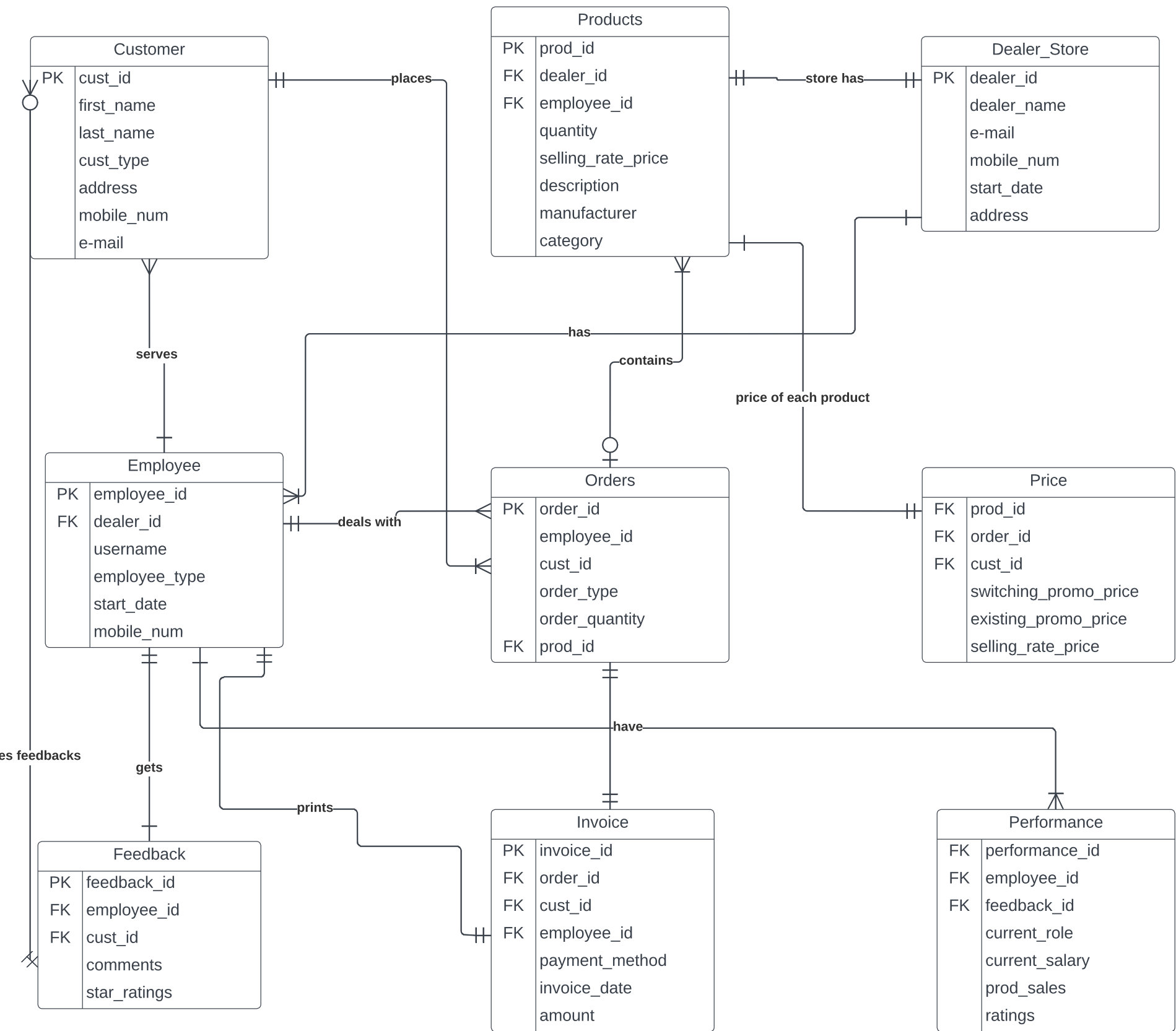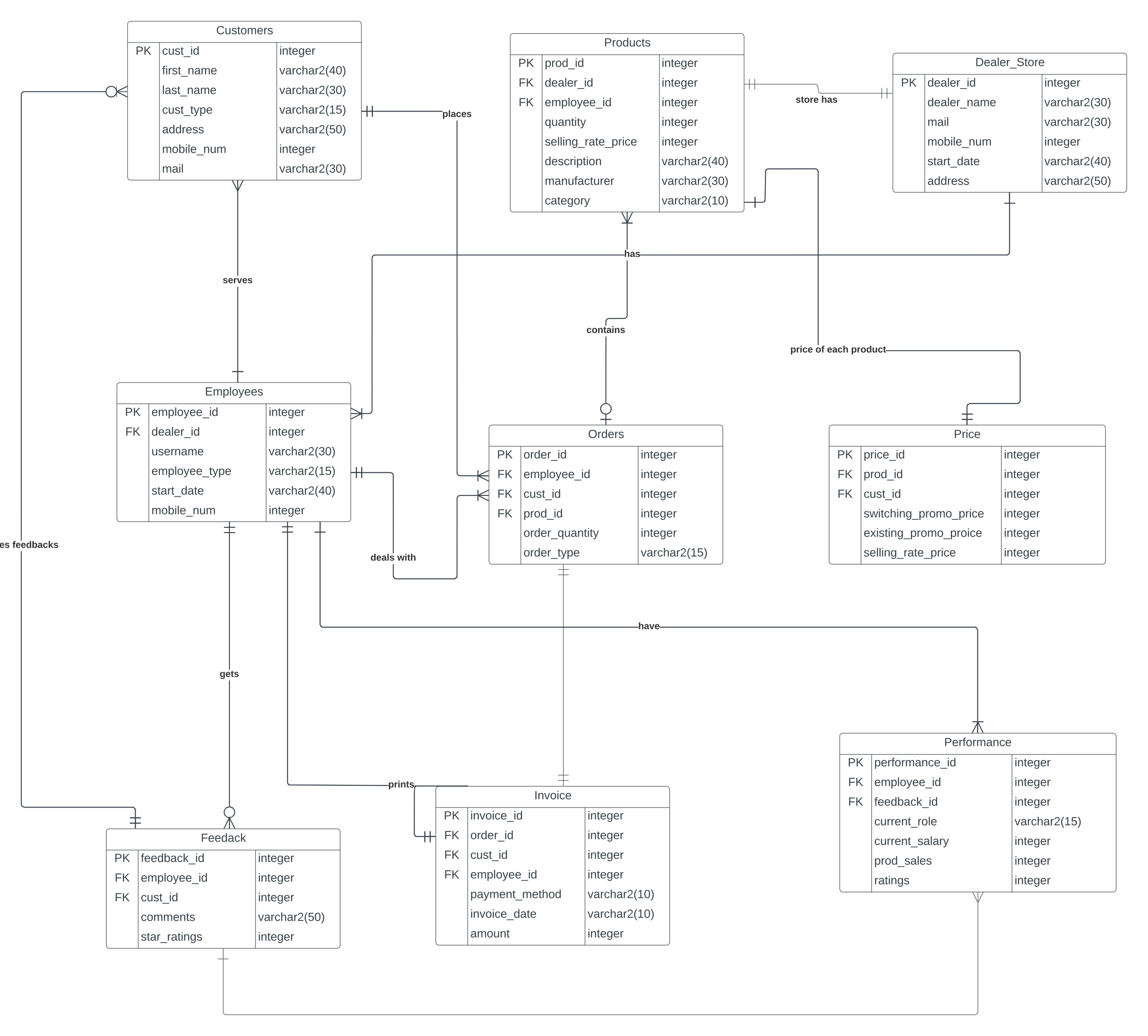
## PROJECT SCOPE:

This project will assist in making the firm function more successfully and methodically so that there won't be any deception. By using data purification, data modeling, and data mining techniques, we can expand this project and discover the insights needed to expand that business. For example, we can perform data mining techniques and explore the data plan table to determine which plan is the most popular and which is the least popular so that we can get insights and make changes to improve customer happiness. Additionally, we can create a dashboard with various charts displaying the sales report, the number of devices, the profit in various months, etc. This analysis will assist the company in growing and meeting the needs of its customers, it helps in improving the firm's performance, and increases its revenue.

| ID | Name |
|------|----------|
| B001 | EC_PAON |
| B002 | EC_AICES |
| B003 | MC_EA |
| B004 | MC_ES |
| B005 | EC_APC |
| B006 | EC_API |
| B007 | EC_CI |
| B008 | EC_SROA |
| B009 | EC_WDS |
| B010 | ECN_MDS |
| B011 | ECN_THI |
| B012 | EFC_TW |
| B013 | ECN_CDCA |
| B014 | ECN_CPO |

**Rule**

Employees can create a customer account, add them in any plan, activate their own old device or new device.

Employees can add the invoice and cannot edit the selling retail price(SRP)

Managers can create the employees account

Managers can edit the selling retail price(SRP)

Employees can apply the promo codes

Employees can add products to the inventory

Employees cannot clock in before 10 AM

Employees can ship or receive products from one dealer store to another dealer store

Employees can work in multiple dealer stores

Employees cannot work in multiple dealer stores on the same day

Employees cannot sell more than 10 Home internet in a day

Employee feedback comment should be less than 20 words

Employees cannot cancel or delete customer account

Employees cannot cancel the pending orders

## Customer
| | |
|---|---|
| PK | cust_id |
| | first_name |
| | last_name |
| | cust_type |
| | address |
| | mobile_num |
| | e-mail |

## Products
| | |
|---|---|
| PK | prod_id |
| FK | dealer_id |
| FK | employee_id |
| | quantity |
| | selling_rate_price |
| | description |
| | manufacturer |
| | category |

## Dealer_Store
| | |
|---|---|
| PK | dealer_id |
| | dealer_name |
| | e-mail |
| | mobile_num |
| | start_date |
| | address |

## Employee
| | |
|---|---|
| PK | employee_id |
| FK | dealer_id |
| | username |
| | employee_type |
| | start_date |
| | mobile_num |

## Orders
| | |
|---|---|
| PK | order_id |
| | employee_id |
| | cust_id |
| | order_type |
| | order_quantity |
| FK | prod_id |

## Price
| | |
|---|---|
| FK | prod_id |
| FK | order_id |
| FK | cust_id |
| | switching_promo_price |
| | existing_promo_price |
| | selling_rate_price |

## Feedback
| | |
|---|---|
| PK | feedback_id |
| FK | employee_id |
| FK | cust_id |
| | comments |
| | star_ratings |

## Invoice
| | |
|---|---|
| PK | invoice_id |
| FK | order_id |
| FK | cust_id |
| FK | employee_id |
| | payment_method |
| | invoice_date |
| | amount |

## Performance
| | |
|---|---|
| FK | performance_id |
| FK | employee_id |
| FK | feedback_id |
| | current_role |
| | current_salary |
| | prod_sales |
| | ratings |

Relationships: places, store has, serves, has, contains, price of each product, deals with, have, gets, prints, gives feedbacks

**Customers**

| PK | cust_id | integer |
|---|---|---|
| | first_name | varchar2(40) |
| | last_name | varchar2(30) |
| | cust_type | varchar2(15) |
| | address | varchar2(50) |
| | mobile_num | integer |
| | mail | varchar2(30) |

**Products**

| PK | prod_id | integer |
|---|---|---|
| FK | dealer_id | integer |
| FK | employee_id | integer |
| | quantity | integer |
| | selling_rate_price | integer |
| | description | varchar2(40) |
| | manufacturer | varchar2(30) |
| | category | varchar2(10) |

**Dealer_Store**

| PK | dealer_id | integer |
|---|---|---|
| | dealer_name | varchar2(30) |
| | mail | varchar2(30) |
| | mobile_num | integer |
| | start_date | varchar2(40) |
| | address | varchar2(50) |

**Employees**

| PK | employee_id | integer |
|---|---|---|
| FK | dealer_id | integer |
| | username | varchar2(30) |
| | employee_type | varchar2(15) |
| | start_date | varchar2(40) |
| | mobile_num | integer |

**Orders**

| PK | order_id | integer |
|---|---|---|
| FK | employee_id | integer |
| FK | cust_id | integer |
| FK | prod_id | integer |
| | order_quantity | integer |
| | order_type | varchar2(15) |

**Price**

| PK | price_id | integer |
|---|---|---|
| FK | prod_id | integer |
| FK | cust_id | integer |
| | switching_promo_price | integer |
| | existing_promo_proice | integer |
| | selling_rate_price | integer |

**Performance**

| PK | performance_id | integer |
|---|---|---|
| FK | employee_id | integer |
| FK | feedback_id | integer |
| | current_role | varchar2(15) |
| | current_salary | integer |
| | prod_sales | integer |
| | ratings | integer |

**Feedack**

| PK | feedback_id | integer |
|---|---|---|
| FK | employee_id | integer |
| FK | cust_id | integer |
| | comments | varchar2(50) |
| | star_ratings | integer |

**Invoice**

| PK | invoice_id | integer |
|---|---|---|
| FK | order_id | integer |
| FK | cust_id | integer |
| FK | employee_id | integer |
| | payment_method | varchar2(10) |
| | invoice_date | varchar2(10) |
| | amount | integer |

Relationships: places, store has, serves, has, contains, price of each product, deals with, have, gets, prints, gives feedbacks

```sql
create table Dealer_Store(
    dealer_id integer generated by default as identity,
    dealer_name varchar2(30) not null,
    mail varchar2(30),
    mobile_num integer not null,
    start_date varchar2(40),
    street_address varchar2(50),
    city varchar2(30),
    country varchar2(30),
    primary key(dealer_id)
);

create table Employees (
    employee_id integer generated by default as identity,
    dealer_id integer not null,
    username varchar2(30) not null,
    employee_type varchar2(30) not null,
    start_date varchar2(40),
    mobile_num integer not null,
    primary key(employee_id),
    foreign key(dealer_id) references dealer_store(dealer_id)
);

create table Customers (
    cust_id integer generated by default as identity,
    employee_id integer not null,
    first_name varchar2(40) not null,
    last_name varchar2(30) not null,
    cust_type varchar2(15) not null,
    street_address varchar2(50),
    city varchar2(30),
    country varchar2(30),
    mobile_num integer not null,
    mail varchar2(30),
    primary key(cust_id),
    foreign key(employee_id) references employees(employee_id)
);

create table Products (
    prod_id integer generated by default as identity,
    dealer_id integer not null,
    quantity integer not null,
    selling_rate_price integer not null,
    category varchar2(40) not null,
    manufacturer varchar2(30) not null,
    primary key(prod_id),
    foreign key(dealer_id) references dealer_store(dealer_id)
);

create table Orders (
    order_id integer generated by default as identity,
    employee_id integer not null,
    cust_id integer not null,
    prod_id integer not null,
    order_quantity integer not null,
```

```sql
    order_type varchar2(15),
    primary key(order_id),
    foreign key(employee_id) references employees(employee_id),
    foreign key(cust_id) references customers(cust_id),
    foreign key(prod_id) references products(prod_id)
);

create table Price (
    price_id integer generated by default as identity,
    prod_id integer not null,
    switching_promo_price integer,
    existing_promo_price integer,
    selling_rate_price integer,
    primary key(price_id),
    foreign key(prod_id) references products(prod_id)
);

create table Invoice (
    invoice_id integer generated by default as identity,
    order_id integer not null,
    cust_id integer not null,
    employee_id integer not null,
    payment_method varchar2(10) not null,
    invoice_date varchar2(10),
    amount integer not null,
    primary key(invoice_id),
    foreign key(order_id) references orders(order_id),
    foreign key(cust_id) references customers(cust_id),
    foreign key(employee_id) references employees(employee_id)
);

create table Feedback (
    feedback_id integer generated by default as identity,
    employee_id integer not null,
    cust_id integer not null,
    comments varchar2(50) not null,
    star_ratings integer not null,
    primary key(feedback_id),
    foreign key(employee_id) references employees(employee_id),
    foreign key(cust_id) references customers(cust_id)
);

create table Performance (
    performance_id integer generated by default as identity,
    employee_id integer not null,
    invoice_id integer not null,
    feedback_id integer not null,
    current_role varchar2(15) not null,
    current_salary integer,
    prod_sales integer,
    ratings integer not null,
    primary key(performance_id),
    foreign key(employee_id) references employees(employee_id),
    foreign key(invoice_id) references invoice(invoice_id),
    foreign key(feedback_id) references feedback(feedback_id)
```

);

```sql
insert into customers
values (1, 'Rhonda', 'Morrison', 'new', null, 'Aurora', 'United States',
'(458) 616-1674', 'cras.eget@aol.net');

insert into customers
values (2, 'Simone', 'Pollard', 'existing', 'Ap #554-6760 Proin Rd.',
'Phoenix', 'United States', '1-668-177-3804', 'ante.lectus@hotmail.com');

insert into customers
values (3, 'Jasper', 'David', 'new', 'Ap #947-7787 Eget St.', null,
'United States', '(778) 416-1551', 'lobortis.augue@outlook.org');

insert into customers
values (4, 'Lionel', 'Drake', 'switcher', '4778 Velit. St.', 'Overland
Park', 'United States', '(368) 431-0634', 'sed.nec@google.org');

insert into customers
values (5, 'Candace', 'Cooper', 'existing', 'Ap #268-1228 Arcu Avenue',
'Columbus', 'United States', '1-776-584-5675', null);

insert into customers
values (6, 'Chaney', 'Rodriquez', 'new', '7753 Est Rd.', 'West Jordan',
'United States', '1-429-786-2761', 'magna.a@aol.com');

insert into customers
values (7, 'Caryn', 'Spence', 'switcher', '331-7935 Arcu. Street',
'Richmond', 'United States', '(757) 333-3626', 'est@hotmail.com');

insert into customers
values (8, 'Meghan', 'Humphrey', 'new', '992-5091 Vitae Street', 'Fort
Smith', 'United States', '(323) 493-1597', null);

insert into customers
values (9, 'Vivian', 'Benjamin', 'switcher', '4326 Ante. Avenue', 'Saint
Paul', 'United States', '1-275-314-4538',
'neque.vitae.semper@google.couk');

insert into customers
values (10, 'Peter', 'Carr', 'existing', 'Ap #915-4049 Dui. St.',
'Montpelier', 'United States', '(261) 453-8684',
'eu.euismod@protonmail.couk');

insert into customers
values (11, 'Lysandra', 'Long', 'existing', 'P.O. Box 760, 5842 Nam Av.',
'Springfield', 'United States', '1-690-715-8034', 'mauris@google.com');

insert into customers
values (12, 'Lamar', 'Gamble', 'new', '203-5178 Nibh Avenue', 'Atlanta',
'United States', '(512) 266-3455', 'est@hotmail.com');

insert into customers
values (13, 'Nichole', 'Bryant', 'switcher', 'P.O. Box 729, 487 Porttitor
Rd.', 'San Jose', 'United States', '(966) 126-3378',
'volutpat@protonmail.org');
```

```sql
insert into customers
values (14, 'Cody', 'Ballard', 'new', '9802 Imperdiet, Road', 'Flint',
'United States', '(612) 876-5028', 'consectetuer.adipiscing@hotmail.net');

insert into customers
values (15, 'Nerea', 'Boyd', 'existing', 'Ap #241-1907 Cum St.',
'Springdale', 'United States', '(741) 140-6661', 'euismod.et@yahoo.couk');

insert into customers
values (16, 'Carlos', 'Keller', 'new', '3796 Quam Avenue', 'Seattle',
'United States', '(494) 820-2096', 'curabitur.ut@icloud.ca');

insert into customers
values (17, 'Paki', 'Irwin', 'new', 'Ap #802-5166 Justo St.', 'New
Orleans', 'United States', '1-382-933-2317', 'sed.pharetra@icloud.com');

insert into customers
values (18, 'Benedict', 'Mcdonald', 'existing', 'Ap #268-9434 Nec Street',
null, 'United States', '(240) 185-1134', 'nec@icloud.couk');

insert into customers
values (19, 'Tate', 'Chandler', 'switcher', '243-6885 Sed St.', 'Baton
Rouge', 'United States', '(553) 538-9833',
'consequat.auctor@hotmail.edu');

insert into customers
values (20, 'Erasmus', 'Hicks', 'existing', '278-7618 Nisi Avenue',
'Chicago', 'United States', '(878) 695-6904', 'vitae@outlook.com');

insert into customers
values (21, 'Kameko', 'Schwartz', 'existing', null, 'Mesa', 'United
States', '(503) 151-5696', 'diam@outlook.net');

insert into customers
values (22, 'Silas', 'Burgess', 'existing', '3397 Sapien. Rd.',
'Sacramento', 'United States', '1-484-813-6536',
'inceptos.hymenaeos.mauris@aol.org');

insert into customers
values (23, 'Tallulah', 'Mckee', 'new', '115-3495 Hendrerit Rd.',
'Oklahoma City', 'United States', '(635) 561-6801',
'elit@protonmail.couk');

insert into customers
values (24, 'Sage', 'Parsons', 'new', 'Ap #575-6139 Nunc. Rd.',
'Honolulu', 'United States', '(275) 154-6579',
'vivamus.non.lorem@protonmail.edu');

insert into customers
values (25, 'Cody', 'Conley', 'switcher', 'Ap #467-9102 Luctus Avenue',
'Miami', 'United States', '1-217-514-7722', 'eu@protonmail.net');

insert into customers
```

```
values (26, 'Sybill', 'Campbell', 'new', 'P.O. Box 171, 6208 Sed St.',
        'Hattiesburg', 'United States', '1-725-141-7081',
'iaculis.enim@aol.net');

insert into customers
values (27, 'Malik', 'Barr', 'existing', '772-4785 Rhoncus. Road',
        'Harrisburg', 'United States', '1-266-633-3183',
'congue@google.org');

insert into customers
values (28, 'Vance', 'Russo', 'existing', 'Ap #126-5422 Viverra. St.',
        'Wichita', 'United States', '1-605-274-3150', null);

insert into customers
values (29, 'Fatima', 'Mccullough', 'switcher', 'Ap #830-9172 In Av.',
        'Birmingham', 'United States', '(670) 840-1484',
'lacus.etiam@aol.com');

insert into customers
values (30, 'Roth', 'Wolf', 'existing', '501-2405 Nibh. Road',
        'Gillette', 'United States', '1-953-558-6338',
'nullam@hotmail.edu');

insert into customers
values (31, 'Lillian', 'Calhoun', 'new', '635-1835 Urna Street',
        'Annapolis', 'United States', '1-743-583-4084',
'tellus.eu@aol.edu');

insert into customers
values (32, 'Violet', 'Lowery', 'switcher', '6763 Id Avenue',
        'Columbia', 'United States', '1-375-252-3250',
'urna.suscipit@aol.ca');

insert into customers
values (33, 'Arden', 'Shepherd', 'new', 'Ap #618-2008 Varius Ave',
        'Aurora', 'United States', '1-564-861-6587', 'ac@icloud.net');

insert into customers
values (34, 'Whilemina', 'Hickman', 'existing', 'P.O. Box 932, 3184
Suspendisse Av.',
        'Las Vegas', 'United States', '1-946-896-7867',
'nibh.sit@aol.com');

insert into customers
values (35,'Jenette', 'Sheppard', 'existing', '185-9979 Sem, Av.',
        'Metairie', 'United States', '(355) 326-8014',
'maecenas@outlook.edu');

insert into customers
values (36,'Bell', 'Collins', 'existing', '8760 Eu Road',
        'Sandy', 'United States', '1-422-485-5470', 'mi@outlook.edu');

insert into customers
values (37, 'Logan', 'Dickerson', 'switcher', '430-9096 At Avenue',
```

```
        'Overland Park', 'United States', '1-954-593-5451',
'vel.est.tempor@outlook.net');

insert into customers
values (38, 'Whitney', 'Byrd', 'existing', '6241 Purus. Avenue',
        'College', 'United States', '(751) 594-6866',
'sed.est@google.ca');

insert into customers
values (39, 'Hedwig', 'Roach', 'new', 'Ap #826-9504 Aliquet. St.',
        null, 'United States', '1-482-610-7782', 'maecenas@outlook.couk');

insert into customers
values (40, 'Holly', 'Mayo', 'existing', '485-9867 Volutpat. Avenue',
        'New Haven', 'United States', '1-787-385-7449',
'neque.non@protonmail.com');

insert into customers
values (41,'Dalton', 'Lowery', 'switcher', 'P.O. Box 822, 9203 Tortor
Avee',
        'Mesa', 'United States', '1-221-483-7498',
'aliquam.ultrices.iaculis@aol.couk');

insert into customers
values (42,'Leila', 'Barron', 'switcher', 'P.O. Box 206, 9352 At, Ave',
        'Athens', 'United States', '(948) 570-7712',
'pretium.aliquet@yahoo.couk');

insert into customers
values (43,'Selma', 'Parker', 'new', '1249 Auctor, St.',
        'Pocatello', 'United States', '1-314-176-1255',
'placerat.eget@aol.edu');

insert into customers
values (44,'Kyra', 'Shepherd', 'existing', '922-7499 Sed Street',
        'Fort Worth', null, '(822) 464-3026', 'vel.convallis@google.com');

insert into customers
values (45,'Hadassah', 'Ward', 'new', 'Ap #410-9070 Mauris St.',
        'Hillsboro', 'United States', '1-341-356-7827',
'fringilla.cursus@hotmail.com');

insert into customers
values (46, 'Zephania', 'Oliver', 'existing', 'Ap #492-6238 Duis Av.',
        'Grand Island', 'United States', '1-507-630-6998',
'ut.erat.sed@protonmail.edu');

insert into customers
values (47, 'Rylee', 'Carson', 'existing', '668-9341 Molestie Rd.',
        'Tampa', 'United States', '1-680-785-5329', 'in@google.edu');

insert into customers
values (48, 'Sonya', 'Downs', 'new', 'Ap #693-5926 Maecenas Rd.',
```

```
        'Springfield', 'United States', '(329) 231-8135',
'pellentesque.habitant@hotmail.com');

insert into customers
values (49,'Yoko', 'Phillips', 'new', 'Ap #496-5464 Malesuada Street',
        'Lewiston', 'United States', '(786) 839-1023',
'non.arcu@google.net');

insert into customers
values (50, 'Byron', 'Mendez', 'switcher', 'Ap #164-3657 Pharetra. St.',
        'Evansville', 'United States', '1-815-636-8831',
'eget@hotmail.org');
```

```sql
insert into dealer_store
values(1, 'sprint','eu.nibh@hotmail.edu', '(663) 169-4288',
'07-09-17', 'Ap #894-4908 Dictum Av.', 'Birmingham', null);

insert into dealer_store
values(2, 'mint',null, '1-979-283-3405',
'11-08-04', 'Ap #574-6442 Tortor. Rd.', 'Kansas City', 'United States');

insert into dealer_store
values(3, 'boost','vivamus@hotmail.com', '1-596-786-6440',
'07-18-03', 'P.O. Box 761, 9288 Sit Road', null, 'United States');

insert into dealer_store
values(4, 'boost', null , '1-664-738-9719',
null, 'Ap #922-6893 Ultrices. Rd.', 'Broken Arrow', null);

insert into dealer_store
values(5, 'mint', 'ornare.libero@outlook.com' , '1-218-617-4051',
'02-26-22', '652-5687 Nisl Ave', 'Lawton', 'United States');

insert into dealer_store
values(6, 'at&t', 'lobortis.nisi.nibh@yahoo.ca' , '1-871-365-8927',
null, null, 'Gillette', 'United States');


insert into dealer_store
values(7, 't-mobile', 'scelerisque.lorem@google.org' , '1-686-661-8226',
'10-14-10', 'Ap #706-8430 Elit Street', 'Worcester', 'United States');

insert into dealer_store
values(8, 'at&t', 'nisl.sem@google.ca' , '(910) 592-4569',
null, '663-7742 Per Rd.', null, 'United States');

insert into dealer_store
values(9, 'xfinity', 'malesuada@yahoo.com' , '1-769-886-6242',
'04-07-12', '417-6767 Venenatis St.', 'Hilo', 'United States');

insert into dealer_store
values(10, 'mint', null , '(531) 321-6399',
'03-27-07', 'P.O. Box 536, 4928 Amet St.', 'Fresno', null);
```

```sql
insert into employees
values(100, 9, 'Oren Hamilton', 'supervisor', null, '(421) 217-7319');

insert into employees
values(101, 8, 'Hiroko Mcgowan', 'cashier', '01-10-03', '1-679-276-9712');

insert into employees
values(102, 4, 'Megan Shepherd', 'manager', '10-25-17', '1-536-745-5661');

insert into employees
values(103, 2, 'Jillian Baker', 'cashier', '11-01-09', '(357) 845-7558');

insert into employees
values(104, 4, 'Kyla Walter', 'cashier', null, '1-320-653-7483');

insert into employees
values(105, 5, 'Kirk Wall', 'cashier', '12-08-17', '1-570-782-4941');

insert into employees
values(106, 1, 'Byron Andrews', 'cashier', '01-24-19', '(196) 461-8981');

insert into employees
values(107, 2, 'Cora Farley', 'manager', null, '1-833-884-8155');

insert into employees
values(108, 10, 'Teagan Gates', 'cashier', '06-02-22', '1-245-643-9728');

insert into employees
values(109, 5, 'Emma Campbell', 'manager', '12-05-21', '1-862-356-3572');

insert into employees
values(110, 8, 'Colton Leonard', 'manager', '03-19-09', '1-911-206-8114');

insert into employees
values(111, 3, 'Talon Flowers', 'supervisor', null, '1-487-480-4392');

insert into employees
values(112, 2, 'Lani Thornton', 'worker', '02-10-05', '1-648-568-3334');

insert into employees
values(113, 10, 'Kathleen Le', 'manager', '05-20-10', '1-823-786-7128');

insert into employees
values(114, 6, 'Brenden Taylor', 'worker', '10-07-09', '1-361-516-2243');

insert into employees
values(115, 9, 'Kasimir William', 'supervisor', '02-09-13', '(127) 631-
2815');

insert into employees
values(116, 1, 'Libby Doyle', 'cashier', '10-22-17', '1-741-950-8754');

insert into employees
values(117, 10, 'Brandon Baker', 'cashier', '04-18-14', '1-932-217-7776');
```

```sql
insert into employees
values(118, 2, 'Mallory Oneil', 'supervisor', '07-17-15', '(813) 447-4757');

insert into employees
values(119, 10, 'Jeanette Mcintyre', 'cashier', '06-02-08', '(796) 343-9029');

insert into employees
values(120, 3, 'Zena Mayer', 'worker', '11-27-08', '(384) 721-6886');

insert into employees
values(121, 10, 'Dora Rodgers', 'worker', '06-13-09', '1-437-479-2578');

insert into employees
values(122, 8, 'Ariel Ellison', 'cashier', '04-10-19', '(527) 248-1686');

insert into employees
values(123, 4, 'Jasper Stone', 'cashier', '12-14-15', '1-703-482-1230');

insert into employees
values(124, 5, 'Genevieve Gentry', 'manager', '04-09-19', '1-585-825-4133');

insert into employees
values(125, 5, 'Brennan Shepherd', 'worker', '08-20-13', '1-181-724-8471');

insert into employees
values(126, 7, 'Nathaniel Valdez', 'manager', '07-11-11', '(249) 511-5573');

insert into employees
values(127, 2, 'Caldwell Brady', 'worker', '10-28-15', '1-832-618-5403');

insert into employees
values(128, 8, 'Leo Christensen', 'cashier', '11-28-13', '(485) 922-7692');

insert into employees
values(129, 4, 'Gavin Oliver', 'worker', '12-10-10', '1-428-767-2358');
```

```sql
INSERT INTO Feedback VALUES(5101, 100, 1, 'Very helpful', 4.8);
INSERT INTO Feedback VALUES(5102, 101, 2, 'Good', 4.0);
INSERT INTO Feedback VALUES(5103, 102, 3, 'Awful service', 1.5);
INSERT INTO Feedback VALUES(5104, 103, 4, 'OK service', 2.5);
INSERT INTO Feedback VALUES(5105, 104, 5, 'Great service', 5);
INSERT INTO Feedback VALUES(5106, 105, 6, 'Average service', 2.5);
INSERT INTO Feedback VALUES(5107, 106, 7, 'Very helpful service', 4.5);
INSERT INTO Feedback VALUES(5108, 107, 8, 'Very helpful service', 4.7);
INSERT INTO Feedback VALUES(5109, 108, 9, 'Very helpful service', 4.8);
INSERT INTO Feedback VALUES(5110, 109, 10, 'Great service', 5);
INSERT INTO Feedback VALUES(5111, 110, 11, 'Very helpful service', 4.8);
INSERT INTO Feedback VALUES(5112, 111, 12, 'Bad service', 2);
INSERT INTO Feedback VALUES(5113, 112, 13, 'Worst service', 1);
INSERT INTO Feedback VALUES(5114, 113, 14, 'Very helpful service', 5);
INSERT INTO Feedback VALUES(5115, 114, 15, 'Average', 3.87);
INSERT INTO Feedback VALUES(5116, 115, 16, 'Very helpful service', 5);
INSERT INTO Feedback VALUES(5117, 116, 17, 'Great service', 4.5);
INSERT INTO Feedback VALUES(5118, 117, 18, 'Average', 3.5);
INSERT INTO Feedback VALUES(5119, 118, 19, 'Very helpful service', 4.8);
INSERT INTO Feedback VALUES(5120, 119, 20, 'Good', 4.1);
INSERT INTO Feedback VALUES(5121, 120, 21, 'Bad', 2.3);
INSERT INTO Feedback VALUES(5122, 121, 22, 'Worst service', 1.5);
INSERT INTO Feedback VALUES(5123, 122, 23, 'Worst service', 1);
INSERT INTO Feedback VALUES(5124, 123, 24, 'Great service', 4.75);
INSERT INTO Feedback VALUES(5125, 124, 25, 'Average', 3);
INSERT INTO Feedback VALUES(5126, 125, 26, 'Very helpful service', 5);
INSERT INTO Feedback VALUES(5127, 126, 27, 'Great', 4.4);
INSERT INTO Feedback VALUES(5128, 127, 28, 'Awful service', 1.5);
INSERT INTO Feedback VALUES(5129, 128, 29, 'Good service', 4.25);
INSERT INTO Feedback VALUES(5130, 129, 30, 'Very helpful service', 4.8);
--repeating employees
INSERT INTO Feedback VALUES(5131, 100, 31, 'Good', 4.25);
INSERT INTO Feedback VALUES(5132, 100, 32, 'Average', 3);
INSERT INTO Feedback VALUES(5133, 101, 33, 'Good', 4.1);
INSERT INTO Feedback VALUES(5134, 102, 34, 'Bad', 2.3);
INSERT INTO Feedback VALUES(5135, 127, 35, 'Very helpful service', 4.5);
INSERT INTO Feedback VALUES(5136, 127, 36, 'Worst service', 1.8);
INSERT INTO Feedback VALUES(5137, 128, 37, 'Very helpful service', 4.5);
INSERT INTO Feedback VALUES(5138, 111, 38, 'Great service', 5);
INSERT INTO Feedback VALUES(5139, 116, 39, 'Worst service', 1.8);
INSERT INTO Feedback VALUES(5140, 119, 40, 'Very helpful service', 4.5);
INSERT INTO Feedback VALUES(5141, 127, 41, 'Average', 3.5);
INSERT INTO Feedback VALUES(5142, 127, 42, 'Average', 3.1);
INSERT INTO Feedback VALUES(5143, 110, 43, 'Great service', 4.75);
INSERT INTO Feedback VALUES(5144, 103, 44, 'Very helpful service', 4.5);
INSERT INTO Feedback VALUES(5145, 101, 45, 'Worst service', 1.8);
INSERT INTO Feedback VALUES(5146, 105, 46, 'Average', 3.5);
INSERT INTO Feedback VALUES(5147, 101, 47, 'Very helpful service', 4.5);
INSERT INTO Feedback VALUES(5148, 128, 48, 'Average', 3.5);
INSERT INTO Feedback VALUES(5149, 115, 49, 'Bad', 2.3);
INSERT INTO Feedback VALUES(5150, 104, 50, 'Worst service', 1);
```

```sql
insert into invoice
values(3, 110, 46, 125, 'cash', '04-27-13', 5345);

insert into invoice
values(16, 120, 25, 109, 'cash', '02-28-17', 8831);

insert into invoice
values(17, 116,35, 109, 'cash', '12-02-19', 1358);

insert into invoice
values(24, 124, 18, 105,'cash','02-18-20', 6200);

insert into invoice
values(26, 147, 12, 110, 'cash', null, 7543);

insert into invoice
values(34, 115, 17, 123, 'cash', '03-15-12', 1918);

insert into invoice
values( 37, 142, 37, 118, 'cash', '11-25-15', 6314);

insert into invoice
values(42, 148, 27, 117,'cash', '12-05-08', 9604);

insert into invoice
values(48, 149, 35, 121, 'cash', '12-15-07', 9713);

insert into invoice
values(49, 149, 35, 129, 'cash', '03-23-19', 2685);

insert into invoice
values(1, 147, 47, 114,'debit', '06-11-07', 5522);

insert into invoice
values(10, 129, 1, 119,'debit', '04-05-07', 4877);

insert into invoice
values(8, 143, 36, 128, 'debit', '07-11-14', 2865);

insert into invoice
values(11, 141, 16, 101, 'debit', '05-23-09', 4985);

insert into invoice
values(12, 135, 14, 124, 'debit', '12-12-16', 2904);

insert into invoice
values(20, 146, 20, 112, 'debit', null, 2708);

insert into invoice
values(21, 115, 11, 123, 'debit', '03-15-19', 4517);

insert into invoice
values(22, 129, 30, 121, 'debit', '08-08-16', 8471);
```

```sql
insert into invoice
values(23, 139, 31, 101, 'debit', '09-28-12', 6107);

insert into invoice
values(28, 107, 18, 115, 'debit', '05-14-14', 4733);

insert into invoice
values(30, 101, 20, 108, 'debit', '02-20-11', 3144);

insert into invoice
values(44, 136, 5, 122, 'debit', '06-05-14', 9262);

insert into invoice
values(46, 143, 43, 122, 'debit', '12-29-07', 9212);

insert into invoice
values(50, 117, 39, 115, 'debit', '08-17-21', 456);

insert into invoice
values (4, 138, 40, 105, 'zelle', '01-10-10', 497);

insert into invoice
values (14, 125, 11, 128, 'zelle', '12-11-11', 8443);

insert into invoice
values (19, 146, 2, 107, 'zelle', '12-03-07', 6065);

insert into invoice
values (25, 106, 1, 101, 'zelle', '03-04-17', 1243);

insert into invoice
values (27, 127, 22, 121, 'zelle', '11-27-09', 2926);

insert into invoice
values (32, 141, 7, 103, 'zelle',null, 5468 );

insert into invoice
values (38, 120, 4, 126, 'zelle', '04-23-13', 1966);

insert into invoice
values (41, 129, 32, 122, 'zelle', '05-16-10', 6311);

insert into invoice
values(2, 109, 24, 109, 'credit', null, 1276);

insert into invoice
values(5, 123, 44, 104, 'credit', '05-25-09', 1282);

insert into invoice
values(29, 129, 48, 115, 'credit', '07-28-11', 7882);

insert into invoice
values(31, 115, 39, 118, 'credit', '09-20-21', 1818);
```

```
insert into invoice
values(33, 101, 37, 122, 'credit', '04-18-10', 6077);

insert into invoice
values(36, 116, 3, 101, 'credit', '08-20-22', 2619);

insert into invoice
values(6, 126, 47, 102, 'venmo', '10-11-12', 1916);

insert into invoice
values(7, 143, 49, 128, 'venmo', '07-13-13', 8803);

insert into invoice
values(9, 124, 38, 100, 'venmo', null, 7535);

insert into invoice
values(13, 112, 12, 101, 'venmo', '03-11-10', 1019);

insert into invoice
values(15, 116, 25, 121, 'venmo', '01-04-06', 2813);

insert into invoice
values(18, 109, 1, 115, 'venmo', '03-12-22', 8820);

insert into invoice
values(35, 140, 44, 113, 'venmo', '12-17-16', 6435);

insert into invoice
values(39, 135, 16, 110, 'venmo', '04-09-12', 719);

insert into invoice
values(40, 114, 4, 116, 'venmo', '06-04-13', 3253);

insert into invoice
values(43, 122, 47, 119, 'venmo', '06-09-19', 1512);

insert into invoice
values(45, 106, 30, 111, 'venmo', '06-09-13', 4909);

insert into invoice
values(47, 121, 14, 106, 'venmo', '02-20-18', 2496);
```

```sql
insert into orders
values(100, 127, 43, 11, 1, 'online');

insert into orders
values(101, 124, 27, 4, 2, 'online');

insert into orders
values(102, 104, 10, 10, 1, null);

insert into orders
values(103, 118, 19, 13, 1, 'online');

insert into orders
values(104, 123, 41, 18, 8, 'online');

insert into orders
values(105, 128, 49, 19, 1, 'online');

insert into orders
values(106, 100, 29, 20, 3, 'online');

insert into orders
values(107, 126, 16, 6, 9, 'online');

insert into orders
values(108, 108, 38, 10, 3, 'in-store');

insert into orders
values(109, 117, 34, 18, 1, 'in-store');

insert into orders
values(110, 109, 4, 16, 4, 'in-store');

insert into orders
values(111, 119, 6, 7, 10, null);

insert into orders
values(112, 103, 20, 14, 10, 'in-store');

insert into orders
values(113, 126, 27, 6, 10, 'online');

insert into orders
values(114, 108, 40, 6, 5, 'in-store');

insert into orders
values(115, 126, 6, 1, 2, 'online');

insert into orders
values(116, 125, 10, 9, 10, 'in-store');

insert into orders
values(117, 106, 1,11, 6, 'in-store');
```

```sql
insert into orders
values(118, 104, 14,16, 2, 'online');

insert into orders
values(119, 119, 15,3, 1, 'in-store');

insert into orders
values(120, 109, 22,18, 3, 'in-store');

insert into orders
values(121, 127, 28,18, 4, 'in-store');

insert into orders
values(122, 116, 41,5, 3, null);

insert into orders
values(123, 117, 21,14, 1, 'in-store');

insert into orders
values(124, 120, 31,14, 2, 'online');

insert into orders
values(125, 117, 26,15, 4, 'in-store');

insert into orders
values(126, 126, 38,9, 4, 'online');

insert into orders
values(127, 110, 24,8, 3, 'online');

insert into orders
values(128, 103, 5,12, 3, 'in-store');

insert into orders
values(129, 120, 23,11, 8, 'in-store');

insert into orders
values(130, 124, 28,2, 8, null);

insert into orders
values(131, 126, 40,19, 3, 'in-store');

insert into orders
values(132, 121, 14,6, 2, 'in-store');

insert into orders
values(133, 106, 35,19, 8, 'in-store');

insert into orders
values(134, 127, 32,9, 7, 'in-store');

insert into orders
values(135, 108, 15,10, 10, 'in-store');
```

```sql
insert into orders
values(136, 120, 24,14, 6, 'in-store');

insert into orders
values(137, 121, 22,10, 7, 'online');

insert into orders
values(138, 127, 37,16, 7, null);

insert into orders
values(139, 109, 14,20, 10, 'in-store');

insert into orders
values(140, 125, 15,3, 1, 'in-store');

insert into orders
values(141, 103, 48,16, 4, 'in-store');

insert into orders
values(142, 103, 1,15, 8, 'online');

insert into orders
values(143, 115, 32,7, 7, 'online');

insert into orders
values(144, 111, 34,4, 2, 'online');

insert into orders
values(145, 129, 45,8, 8, 'online');

insert into orders
values(146, 100, 2,7, 8, 'online');

insert into orders
values(147, 121,50,14, 8, 'in-store');

insert into orders
values(148, 109,47,1, 4, 'online');

insert into orders
values(149, 102,1,7, 9, 'in-store');
```

```sql
insert into performance
values(1001, 103, 5101, 'Manager', 2641, 82, 9.8);

insert into performance
values(1002, 114, 5102, 'Manager', null, 94, 8);

insert into performance
values(1003, 125, 5103, 'Supervisor', 4445, 70, 3.5);

insert into performance
values(1004, 103, 5104, 'Supervisor', 3679, null, 3.5);

insert into performance
values(1005, 129, 5105, 'Cashier', 213, 65, 9);

insert into performance
values(1006, 105, 5106, 'Supervisor', 765, 16, 5.5);

insert into performance
values(1007, 122, 5107, 'Cashier', 3443, 17, 7.5);

insert into performance
values(1008, 103, 5108, 'Manager', 4533, null, 7.7);

insert into performance
values(1009, 103, 5109, 'Cashier', 4093, 73, 9.8);

insert into performance
values(1010, 102, 5110, 'Worker', 4662, 2, 8);

insert into performance
values(1011, 109, 5111, 'Manager', 2206, 17, 8);

insert into performance
values(1012, 102, 5112, 'Cashier', 742, 11, 3);

insert into performance
values(1013, 10, 5113, 'Worker', 721, 13, 2);

insert into performance
values(1014, 114, 5114, 'Cashier', 4666, 81, 10);

insert into performance
values(1015, 125, 5115, 'Cashier', 948, null, 5.87);

insert into performance
values(1016, 103, 5116, 'Supervisor', 897, 6, 8);

insert into performance
values(1017, 129, 5117, 'Cashier', 4195, 8, 4.5);

insert into performance
values(1018, 105, 5118, 'Supervisor', 2128, 32, 5.5);
```

```
insert into performance
values(1019, 122, 5119, 'Cashier', 3542, 78, 9.8);

insert into performance
values(1020, 103, 5120, 'Worker', 1525, null, 5.1);

insert into performance
values(1021, 103, 5121, 'Worker', 4373, 31, 3.3);

insert into performance
values(1022, 109, 5122, 'Supervisor', 4685, 10, 2.5);

insert into performance
values(1023, 102, 5123, 'Supervisor', 3242, 85, 3);

insert into performance
values(1024, 119, 5124, 'Worker', 400, 17, 7.75);

insert into performance
values(1025, 102, 5125, 'Worker', 4770, 69, 6);

insert into performance
values(1026, 104, 5126, 'Worker', 1093, 37, 8.4);

insert into performance
values(1027, 124, 5127, 'Supervisor', 2437, 32, 7.4);

insert into performance
values(1028, 108, 5128, 'Supervisor', 1992, 75, 3.5);

insert into performance
values(1029, 122, 5129, 'Manager', 2749, 36, 5.25);

insert into performance
values(1030, 102, 5130, 'Manager', 4407, 24, 7.8);
```

```
insert into price
values(1,20,150, 200, 180);

insert into price
values(2, 18, null, 400, 250);

insert into price
values(3, 4, 140, 200, 130);

insert into price
values(4, 5, 10, 10, 10);

insert into price
values(5, 13, null, 850, 750);

insert into price
values(6, 17, 300, 400, null);

insert into price
values(7, 8, 180, 300, 250);

insert into price
values(8, 6, 700, 850, 750);

insert into price
values(9, 15, 100, 150, 125);

insert into price
values(10, 1, 200, 300, 150);

insert into price
values(11, 2, 100, 150, 125);

insert into price
values(12, 10, 750, 800, 780);

insert into price
values(13, 19, 200, 300, 275);

insert into price
values(14, 16, 200, 300, 150);

insert into price
values(15, 11, 130, 150, 100);

insert into price
values(16, 9, 300, 400, 340);

insert into price
values(17, 7, 140, 200, 175);

insert into price
values(18, 14, 175, 200, 180);
```

```
insert into price
values(19, 12, 300, 400, 375);

insert into price
values(20, 3, 150, 200, 175);
```

```sql
insert into Products (dealer_id, quantity, category, selling_rate_price,
manufacturer)
values(2, 1, 'adapters', 300, 'samsung');

insert into Products (dealer_id, quantity, category, selling_rate_price,
manufacturer)
values(7, 8, 'mobile', 200, 'samsung');

insert into Products (dealer_id, quantity, category, selling_rate_price,
manufacturer)
values(5, 6, 'mobile', 150, 'oneplus');

insert into Products (dealer_id, quantity, category, selling_rate_price,
manufacturer)
values(4, 7, 'simcard', 200, 'apple');

insert into Products (dealer_id, quantity, category, selling_rate_price,
manufacturer)
values(1, 10, 'headset', 10, 'apple');

insert into Products (dealer_id, quantity, category, selling_rate_price,
manufacturer)
values(7, 4, 'adapters', 850, 'oneplus');

insert into Products (dealer_id, quantity, category, selling_rate_price,
manufacturer)
values(9, 3, 'adapters', 200, 'samsung');

insert into Products (dealer_id, quantity, category, selling_rate_price,
manufacturer)
values(2, 3, 'adapters', 300, 'samsung');

insert into Products (dealer_id, quantity, category, selling_rate_price,
manufacturer)
values(2, 2, 'mobile', 400, 'oneplus');

insert into Products (dealer_id, quantity, category, selling_rate_price,
manufacturer)
values(6, 5, 'headset', 850, 'apple');

insert into Products (dealer_id, quantity, category, selling_rate_price,
manufacturer)
values(3, 6, 'simcard', 150, 'oneplus');

insert into Products (dealer_id, quantity, category, selling_rate_price,
manufacturer)
values(3, 9, 'speakers', 400, 'apple');

insert into Products (dealer_id, quantity, category, selling_rate_price,
manufacturer)
values(5, 2, 'speakers', 850, 'apple');

insert into Products (dealer_id, quantity, category, selling_rate_price,
manufacturer)
```

```sql
values(1, 8, 'speakers', 200, 'oneplus');

insert into Products (dealer_id, quantity, category, selling_rate_price,
manufacturer)
values(8, 7, 'simcard', 150, 'oneplus');

insert into Products (dealer_id, quantity, category, selling_rate_price,
manufacturer)
values(1, 10, 'simcard', 300, 'apple');

insert into Products (dealer_id, quantity, category, selling_rate_price,
manufacturer)
values(1, 2, 'speakers', 400, 'samsung');

insert into Products (dealer_id, quantity, category, selling_rate_price,
manufacturer)
values(3, 7, 'simcard', 400, 'oneplus');

insert into Products (dealer_id, quantity, category, selling_rate_price,
manufacturer)
values(10, 10, 'headset', 300, 'apple');

insert into Products (dealer_id, quantity, category, selling_rate_price,
manufacturer)
values(4, 8, 'speakers', 200, 'apple');
```

```sql
ALTER TABLE employees
rename column start_date to start_date1;

ALTER TABLE employees
ADD (start_date DATE);

UPDATE employees
SET start_date = TO_DATE(start_date1,'MM-DD-YY');

ALTER TABLE employees
DROP (start_date1);

select * from employees;

ALTER TABLE invoice
rename column invoice_date to start_date1;

ALTER TABLE invoice
ADD (invoice_date DATE);

UPDATE invoice
SET invoice_date = TO_DATE(start_date1,'MM-DD-YY');

ALTER TABLE invoice
DROP (start_date1);

ALTER TABLE dealer_store
rename column start_date to start_date1;

ALTER TABLE dealer_store
ADD (start_date DATE);

UPDATE dealer_store
SET start_date = TO_DATE(start_date1,'MM-DD-YY');

ALTER TABLE dealer_store
DROP (start_date1);

select * from dealer_store;
```

# 1. Finding out which employees are working for at&t store:

```
SELECT
    e.employee_id,
    e.username,
    e.mobile_num,
    e.start_date,
    d.dealer_id,
    d.dealer_name
FROM employees e
INNER JOIN dealer_store d
ON e.dealer_id = d.dealer_id
WHERE d.dealer_name = 'at&t'
ORDER BY e.employee_id;
```

**Output:**

| EMPLOYEE_ID | USERNAME | MOBILE_NUM | START_DATE | DEALER_ID | DEALER_NAME |
|---|---|---|---|---|---|
| 101 | Hiroko Mcgowan | 1-679-276-9712 | 10-JAN-03 | 8 | at&t |
| 110 | Colton Leonard | 1-911-206-8114 | 19-MAR-09 | 8 | at&t |
| 114 | Brenden Taylor | 1-361-516-2243 | 07-OCT-09 | 6 | at&t |
| 122 | Ariel Ellison | (527) 248-1686 | 10-APR-19 | 8 | at&t |
| 128 | Leo Christensen | (485) 922-7692 | 28-NOV-13 | 8 | at&t |

# 2. Finding all the records of products and orders:

```
SELECT
    p.category,
    p.manufacturer,
    o.order_quantity,
    o.order_type
FROM products p
FULL OUTER JOIN
orders o
USING (prod_id)
```

**Output:**

| CATEGORY | MANUFACTURER | ORDER_QUANTITY | ORDER_TYPE |
|----------|--------------|----------------|------------|
| simcard | oneplus | 1 | online |
| simcard | apple | 2 | online |
| headset | apple | 1 | - |
| speakers | apple | 1 | online |
| simcard | oneplus | 8 | online |
| headset | apple | 1 | online |
| speakers | apple | 3 | online |
| adapters | oneplus | 9 | online |
| headset | apple | 3 | in-store |
| simcard | oneplus | 1 | in-store |
| simcard | apple | 4 | in-store |
| adapters | samsung | 10 | - |
| speakers | oneplus | 10 | in-store |
| adapters | oneplus | 10 | online |
| adapters | oneplus | 5 | in-store |
| adapters | samsung | 2 | online |
| mobile | oneplus | 10 | in-store |
| simcard | oneplus | 6 | in-store |
| simcard | apple | 2 | online |

## 3. Get the count of each payment method used for payment over $500 when count is greater than 5.

SELECT payment_method, COUNT(*) AS count
FROM invoice
WHERE amount > 500
GROUP BY payment_method
HAVING COUNT(*) > 5;

**Output:**

| PAYMENT_METHOD | COUNT |
|----------------|-------|
| cash | 10 |
| credit | 6 |
| zelle | 7 |
| venmo | 12 |
| debit | 13 |

**4. Find average star rating of every different comment.**

SELECT DISTINCT comments,
        ROUND(AVG(star_ratings)) as Average_Star_Ratings
FROM feedback
GROUP BY comments;

| COMMENTS | AVERAGE_STAR_RATINGS |
|---|---|
| Good | 4 |
| OK service | 3 |
| Great | 4 |
| Bad | 2 |
| Good service | 4 |
| Worst service | 1 |
| Very helpful | 5 |
| Awful service | 2 |
| Average service | 3 |
| Great service | 5 |
| Very helpful service | 5 |
| Bad service | 2 |
| Average | 3 |

**5. Obtain the information of each customer who has given the employee at least two stars, including their first and last names, comments, and star ratings, and rank the customers according to their star ratings, with the highest star ratings appearing first.**

SELECT
   c.cust_id,
   c.first_name,
   c.last_name,
   f.comments,
   f.star_ratings,
   DENSE_RANK() OVER (ORDER BY f.star_ratings DESC) as ranking
FROM customers c

INNER JOIN feedback f
ON c.cust_id = f.cust_id
WHERE f.star_ratings >= 2;
**Output:**

| CUST_ID | FIRST_NAME | LAST_NAME | COMMENTS | STAR_RATINGS | RANKING |
|---------|-----------|-----------|----------|--------------|---------|
| 16 | Carlos | Keller | Very helpful service | 5 | 1 |
| 14 | Cody | Ballard | Very helpful service | 5 | 1 |
| 10 | Peter | Carr | Great service | 5 | 1 |
| 26 | Sybill | Campbell | Very helpful service | 5 | 1 |
| 38 | Whitney | Byrd | Great service | 5 | 1 |
| 5 | Candace | Cooper | Great service | 5 | 1 |
| 11 | Lysandra | Long | Very helpful service | 4.8 | 2 |
| 30 | Roth | Wolf | Very helpful service | 4.8 | 2 |
| 1 | Rhonda | Morrison | Very helpful | 4.8 | 2 |
| 19 | Tate | Chandler | Very helpful service | 4.8 | 2 |
| 9 | Vivian | Benjamin | Very helpful service | 4.8 | 2 |
| 43 | Selma | Parker | Great service | 4.75 | 3 |
| 24 | Sage | Parsons | Great service | 4.75 | 3 |
| 8 | Meghan | Humphrey | Very helpful service | 4.7 | 4 |
| 40 | Holly | Mayo | Very helpful service | 4.5 | 5 |
| 17 | Paki | Irwin | Great service | 4.5 | 5 |
| 7 | Caryn | Spence | Very helpful service | 4.5 | 5 |
| 44 | Kyra | Shepherd | Very helpful service | 4.5 | 5 |
| 47 | Rylee | Carson | Very helpful service | 4.5 | 5 |

**6. Get average selling price, maximum selling price, minimum selling price of every comapany's various products.**

SELECT
    manufacturer,
    category,
    TRUNC(AVG(selling_rate_price)) AS average_SRP,
    MAX(selling_rate_price) AS max_SRP,
    MIN(selling_rate_price) AS min_SRP
FROM products
WHERE quantity <> 0
GROUP BY manufacturer, category;

**Output:**

| MANUFACTURER | CATEGORY | AVERAGE_SRP | MAX_SRP | MIN_SRP |
|---|---|---|---|---|
| oneplus | adapters | 850 | 850 | 850 |
| apple | speakers | 483 | 850 | 200 |
| oneplus | mobile | 275 | 400 | 150 |
| apple | simcard | 250 | 300 | 200 |
| samsung | mobile | 200 | 200 | 200 |
| oneplus | simcard | 233 | 400 | 150 |
| samsung | speakers | 400 | 400 | 400 |
| samsung | adapters | 266 | 300 | 200 |
| apple | headset | 386 | 850 | 10 |
| oneplus | speakers | 200 | 200 | 200 |

**7. Select customer id, their full name, their type, their order quantity, and the area code. We only need new and existing customers.**

```
SELECT
    c.cust_id,
    (c.first_name || ' '  || c.last_name) AS full_name,
    c.cust_type,
    o.order_quantity,
    (CASE
        WHEN c.mobile_num LIKE '(_)%' THEN RTRIM(LTRIM(SUBSTR(c.mobile_num, 1, 5), '('), ')')
        WHEN c.mobile_num LIKE '1-%' THEN SUBSTR(c.mobile_num, 3, 3)
     ELSE NULL END) AS area_code
FROM customers c
INNER JOIN orders o
ON c.cust_id = o.cust_id
WHERE o.order_type = 'online' AND c.cust_type IN ('existing', 'new');
```

**Output:**

| CUST_ID | FULL_NAME | CUST_TYPE | ORDER_QUANTITY | AREA_CODE |
|---------|-----------|-----------|----------------|-----------|
| 1 | Rhonda Morrison | new | 8 | 458 |
| 2 | Simone Pollard | existing | 8 | 668 |
| 6 | Chaney Rodriquez | new | 2 | 429 |
| 14 | Cody Ballard | new | 2 | 612 |
| 16 | Carlos Keller | new | 9 | 494 |
| 22 | Silas Burgess | existing | 7 | 484 |
| 24 | Sage Parsons | new | 3 | 275 |
| 27 | Malik Barr | existing | 2 | 266 |
| 27 | Malik Barr | existing | 10 | 266 |
| 31 | Lillian Calhoun | new | 2 | 743 |
| 34 | Whilemina Hickman | existing | 2 | 946 |
| 38 | Whitney Byrd | existing | 4 | 751 |
| 43 | Selma Parker | new | 1 | 314 |
| 45 | Hadassah Ward | new | 8 | 341 |
| 47 | Rylee Carson | existing | 4 | 680 |
| 49 | Yoko Phillips | new | 1 | 786 |

**8. Selecting only first 10 employee username, employee type, star ratings from feedback and percent ranking by arranging the star ratings in ascending order:**

SELECT
   E.username,
   E.employee_type,
   star_ratings,
   ROUND(PERCENT_RANK() OVER (PARTITION BY E.employee_type ORDER BY UP.star_ratings), 4)
Rankings
FROM (SELECT employee_id,
        star_ratings
    FROM feedback
    GROUP BY employee_id, star_ratings
    HAVING star_ratings < 3 )UP, employees E
WHERE UP.employee_id = E.employee_id
GROUP BY E.username, E.employee_type, star_ratings;
FETCH FIRST 10 ROWS ONLY;

**Output:**

| USERNAME | EMPLOYEE_TYPE | STAR_RATINGS | RANKINGS |
|----------|---------------|--------------|----------|
| Ariel Ellison | cashier | 1 | 0 |
| Kyla Walter | cashier | 1 | 0 |
| Hiroko Mcgowan | cashier | 1.8 | .4 |
| Libby Doyle | cashier | 1.8 | .4 |
| Jillian Baker | cashier | 2.5 | .8 |
| Kirk Wall | cashier | 2.5 | .8 |
| Megan Shepherd | manager | 1.5 | 0 |
| Megan Shepherd | manager | 2.3 | 1 |
| Talon Flowers | supervisor | 2 | 0 |
| Kasimir William | supervisor | 2.3 | 1 |
| Lani Thornton | worker | 1 | 0 |
| Caldwell Brady | worker | 1.5 | .25 |
| Dora Rodgers | worker | 1.5 | .25 |
| Caldwell Brady | worker | 1.8 | .75 |
| Zena Mayer | worker | 2.3 | 1 |

*9. Selecting invoice id, payment method, employee username, order id by joining invoice table and employees table where the invoice_date is greater than 1st Jan 2006 and less than 9th December 2020, whose customer's first name is Cody and customer's last name is Conley, order type is 'in-store', quantity is equal to 3 and employee id is equal to 109:*

SELECT
  i.invoice_id,
  i.payment_method,
  e.username,
  i.order_id
FROM invoice i
LEFT OUTER JOIN employees e
ON i.employee_id = e.employee_id
WHERE i.invoice_date > '01-JAN-2006' and i.invoice_date < '09-DEC-2020'
AND i.cust_id = (SELECT cust_id
      FROM customers
      WHERE first_name = 'Cody' AND last_name = 'Conley')
AND i.order_id = (SELECT order_id

FROM orders
WHERE order_type = 'in-store' AND order_quantity = 3 AND employee_id = 109);

**Output:**

| INVOICE_ID | PAYMENT_METHOD | USERNAME | ORDER_ID |
|---|---|---|---|
| 16 | cash | Emma Campbell | 120 |