# CUSTOMER RELATIONSHIP MANAGEMENT SYSTEM

# IST 659: DATABASE ADMINISTRATION CONCEPTS AND DATABASE MANAGEMENT SYSTEM

## Team: Data GenNext

| | |
|---|---|
| Amesh Gharat | 414922819 |
| Aditya Shah | 747686910 |
| Manan Vora | 349846253 |

# Table of Contents

# 1) Business Problem

Companies need to keep a track of their customers and customer activity. The employees of the company also need to keep tabs on their clients to make sure that all the production and financial needs of their clients are fulfilled. They also need to know which product was delivered to a particular client, through a salesman and at what price did the client purchased that item.

The company will also need to track the performance of their salesman considering the number of sales that the employee makes in a particular amount of time and the satisfaction of his/her clients. The employees also need to keep a track of the tasks they need to perform.

# 2) Suggested Solution

The solution that Data GenNext has come up with to solve the above business problem is a database management system for the company in the form of a Customer Relationship Management System (CRM). The CRM was designed using MS SQL. This database consists of all the salient information about the employees (salesperson), orders, accounts, products, and tasks supposed to be done by employees. The database management system allows the administrator to store all the information about the above salient features. Also, the employees save up a lot of time as they can just access the CRM interface to keep track of the activities and requirements of the customers.

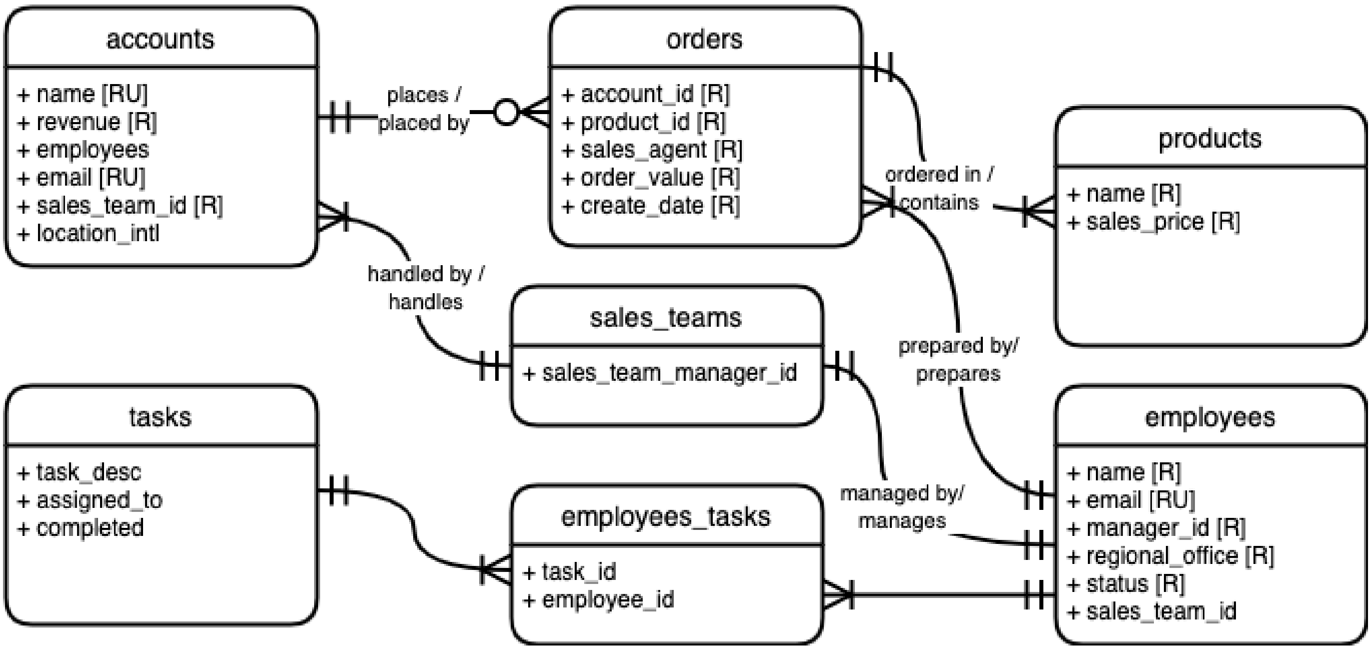
## 3) Topic Selection

We selected the topic of Customer Relationship Management System as there has been a massive overhaul in the operations and functioning of many companies due to the global COVID-19 pandemic. More and more companies are shifting to remote working models and the in-person interactions between customers and their employees have become very unsafe and subsequently rare. Employees have started working remotely and almost the entire work has been shifted over to cloud-based systems. So, our team at Data GenNext has decided to design and implement a database management system in the form of a CRM. With the help of this CRM, customers and employees can interact with each other remotely without the necessity of meeting in person. In the same way, the employees of the company can keep track of their product dispatch, client requests, orders, tasks, and performance remotely. In this way, this CRM will allow interactions between employees and clients remotely with the risk of disease transmission. The main reason for the selection of this project was that we recognized the necessity and convenience of remote engagement.

# 4) Identification of Data Logic

Throughout the project, we have implemented data logic like Procedures and Transactions. The procedures will be simplifying the process of denoting the completion of tasks in the database and assigning new tasks based on client requirements. Transactions are used to insert data into the tables. They are particularly helpful when inserting large amounts of data in one attempt. If there is an error in inserting any entry, the database rolls back to its original state without adding any more entries. We have also used joins to display multiple tables. The code for this logic can be found in the Script of Database section.

# 5) Conceptual Data Model



PRIMARY KEY – FOREIGN KEY CONSTRAINTS IN THE TABLES:

1. Accounts:

    PK – account_id FK – sales_team_id

2. Orders:

    PK - order_id FK - account_id, product_id, sales_agent_id

3. Products:

    PK – product_id

4. Tasks:

    PK - task_id

5. Sales_Teams:

    PK – sales_team _id

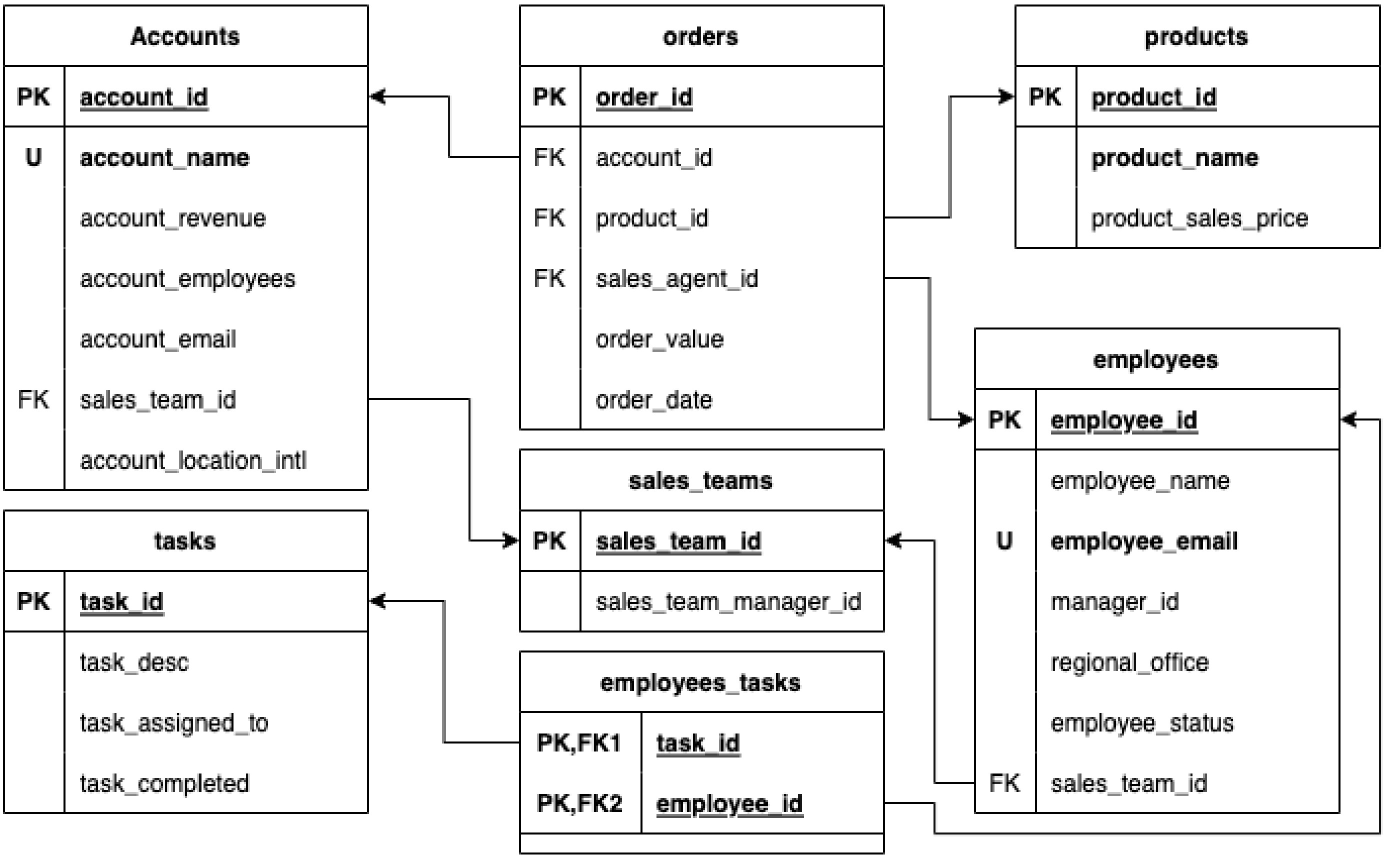
6. Employees:

    PK - employee_id FK - sales_team_id

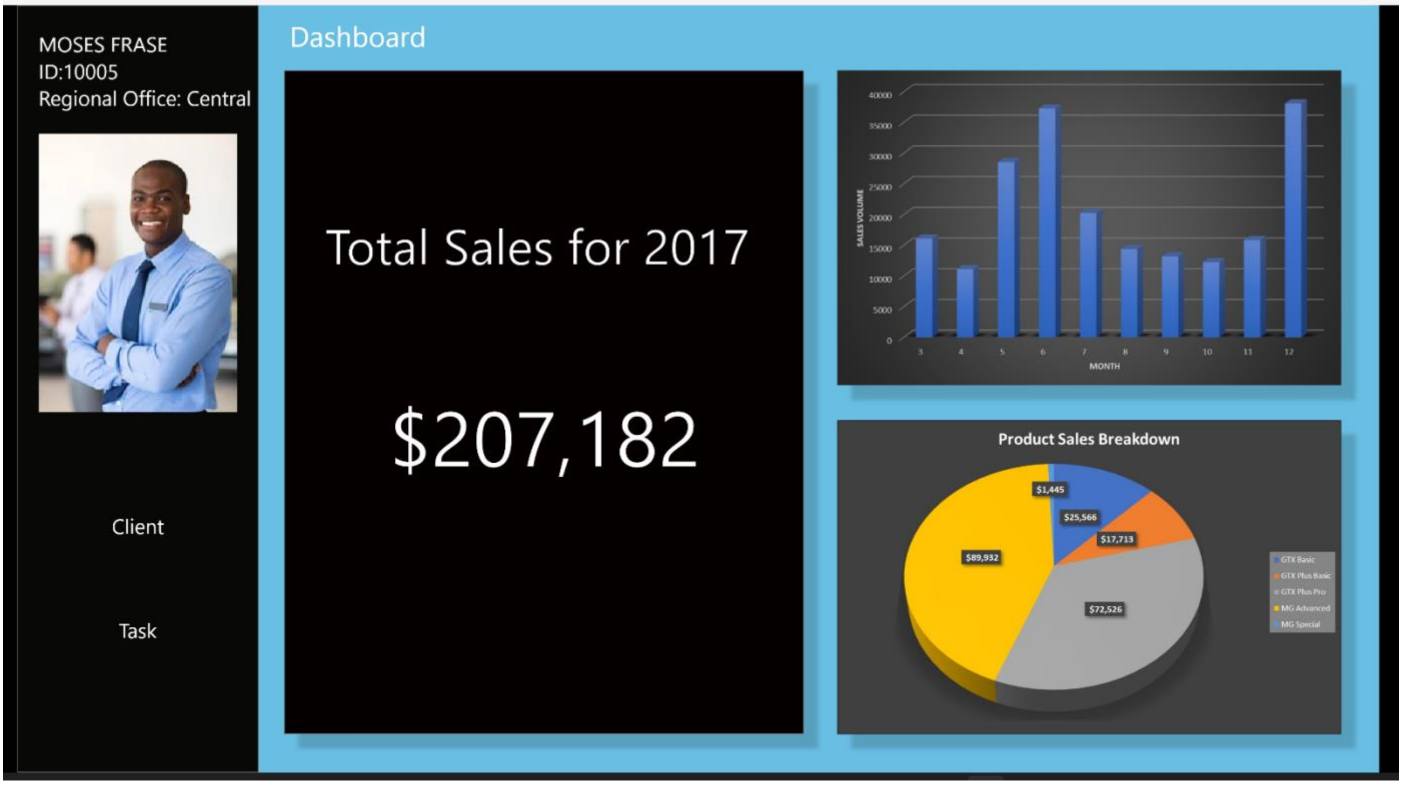
7. Employee_Tasks:

    PK – task_id FK – employee_id

# 6) Logical Data Model



# 7) Application Screens

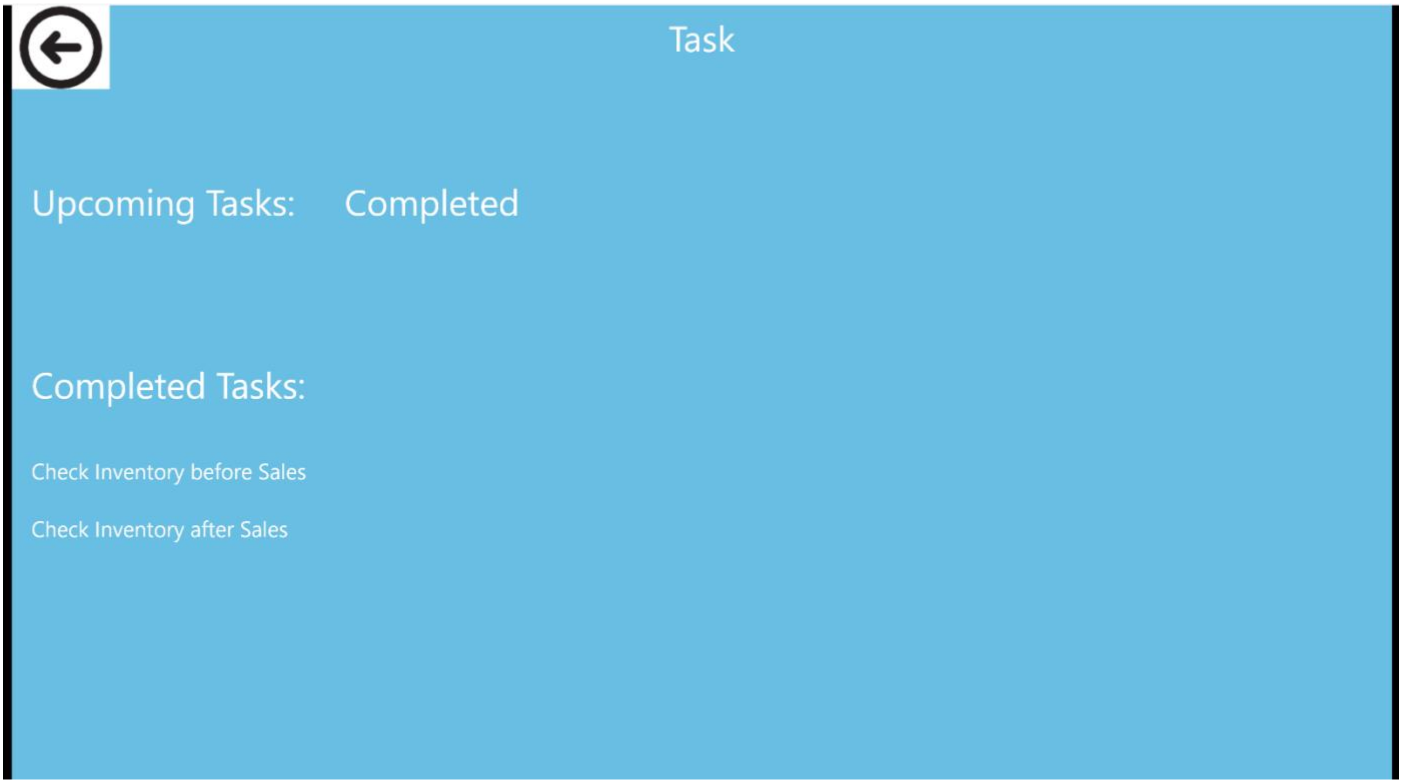


Dashboard

MOSES FRASE
ID:10005
Regional Office: Central

Total Sales for 2017

$207,182



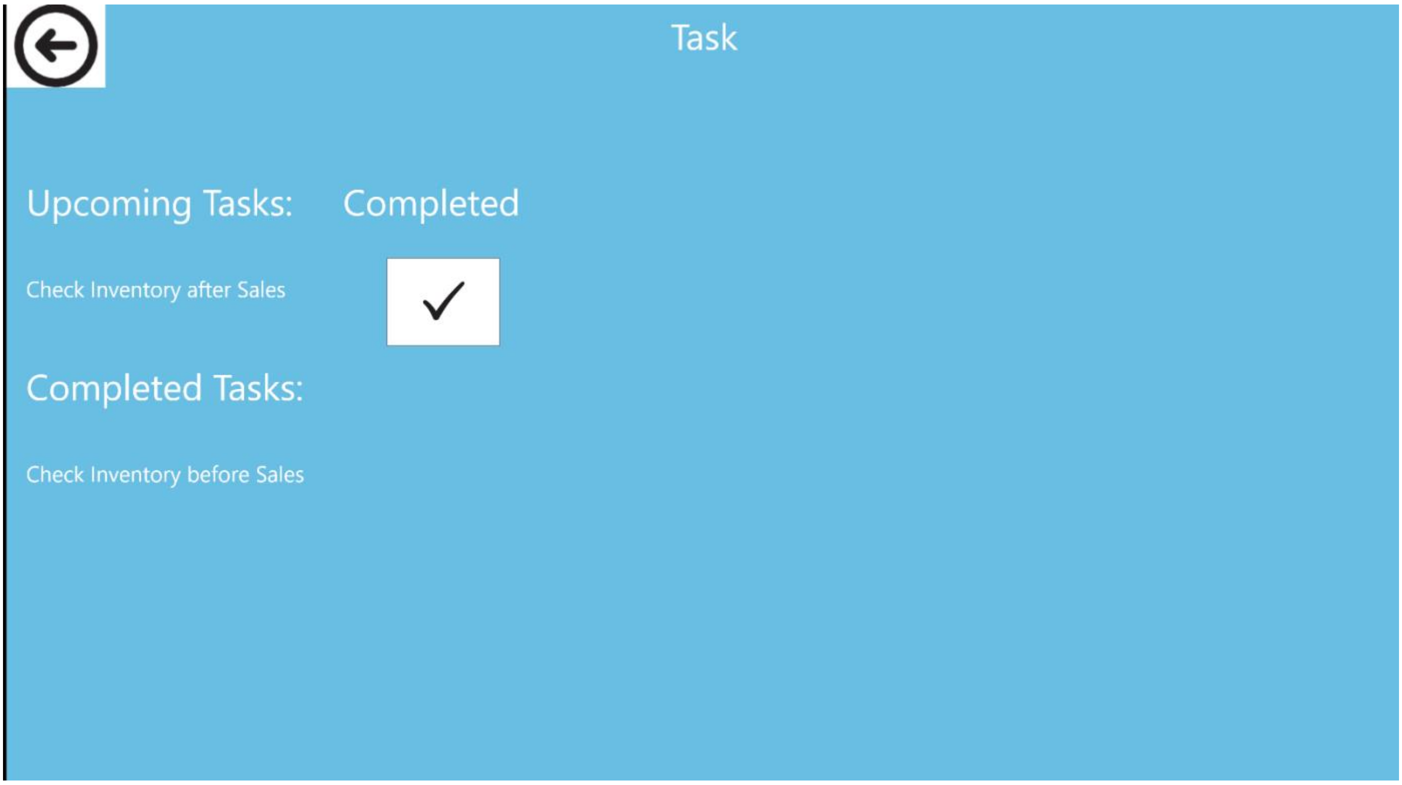
Task

Upcoming Tasks: Completed

Completed Tasks:

Check Inventory before Sales

Check Inventory after Sales



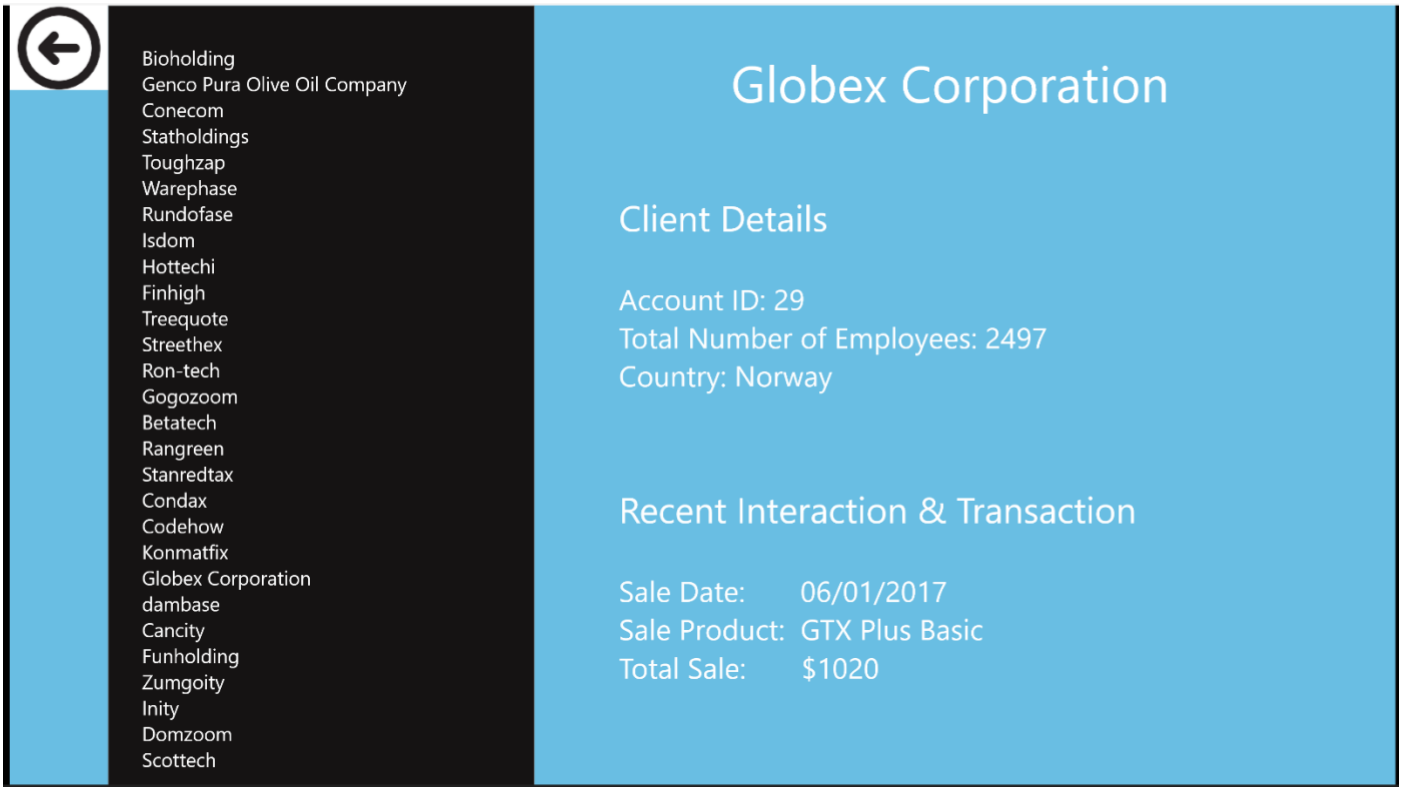
Task

Upcoming Tasks: Completed

Check Inventory after Sales ✓

Completed Tasks:

Check Inventory before Sales



Bioholding
Genco Pura Olive Oil Company
Conecom
Statholdings
Toughzap
Warephase
Rundofase
Isdom
Hottechi
Finhigh
Treequote
Streethex
Ron-tech
Gogozoom
Betatech
Rangreen
Stanredtax
Condax
Codehow
Konmatfix
Globex Corporation
dambase
Cancity
Funholding
Zumgoity
Inity
Domzoom
Scottech

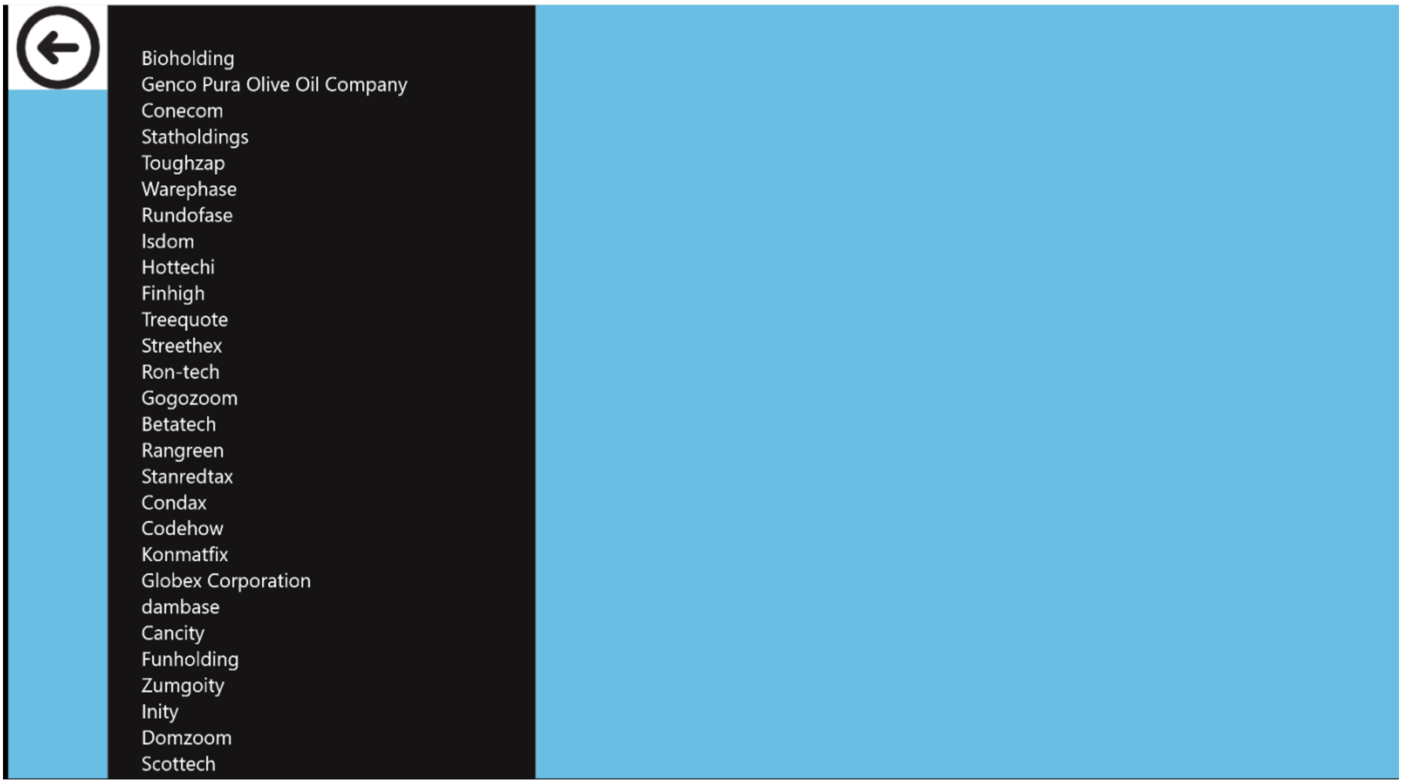
Globex Corporation

Client Details

Account ID: 29
Total Number of Employees: 2497
Country: Norway

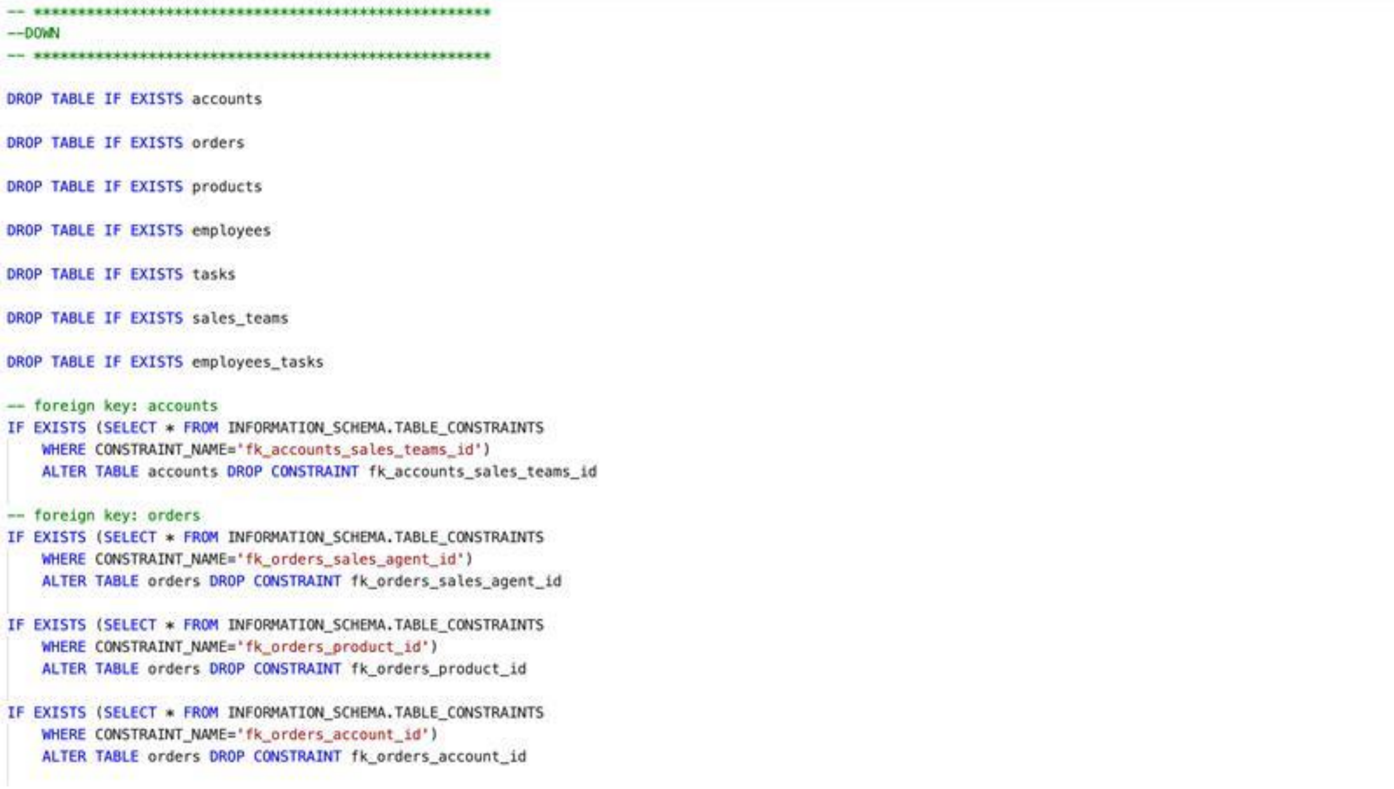
Recent Interaction & Transaction

| | |
|---|---|
| Sale Date: | 06/01/2017 |
| Sale Product: | GTX Plus Basic |
| Total Sale: | $1020 |

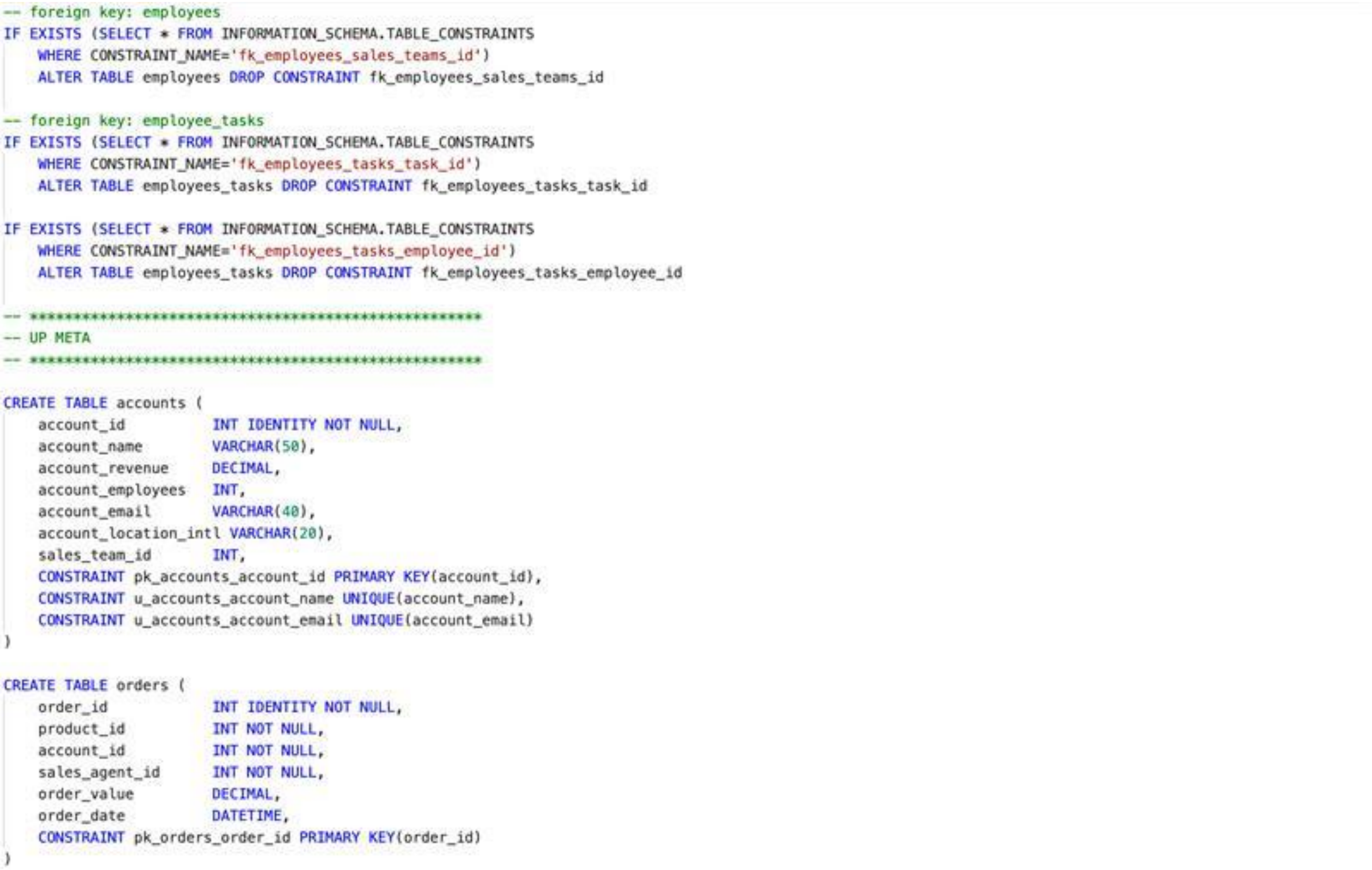


Bioholding
Genco Pura Olive Oil Company
Conecom
Statholdings
Toughzap
Warephase
Rundofase
Isdom
Hottechi
Finhigh
Treequote
Streethex
Ron-tech
Gogozoom
Betatech
Rangreen
Stanredtax
Condax
Codehow
Konmatfix
Globex Corporation
dambase
Cancity
Funholding
Zumgoity
Inity
Domzoom
Scottech

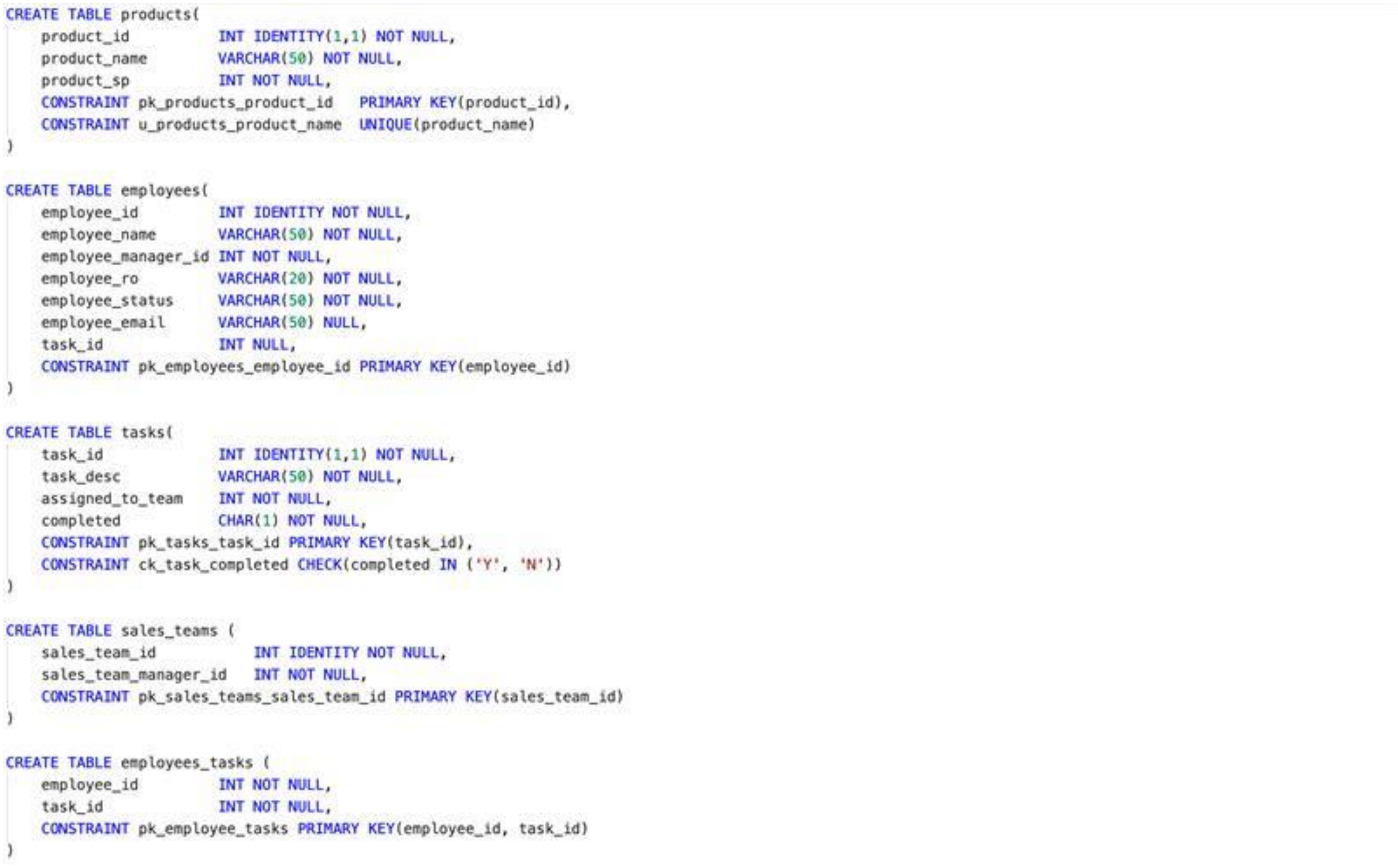
# 8) Script for the Database



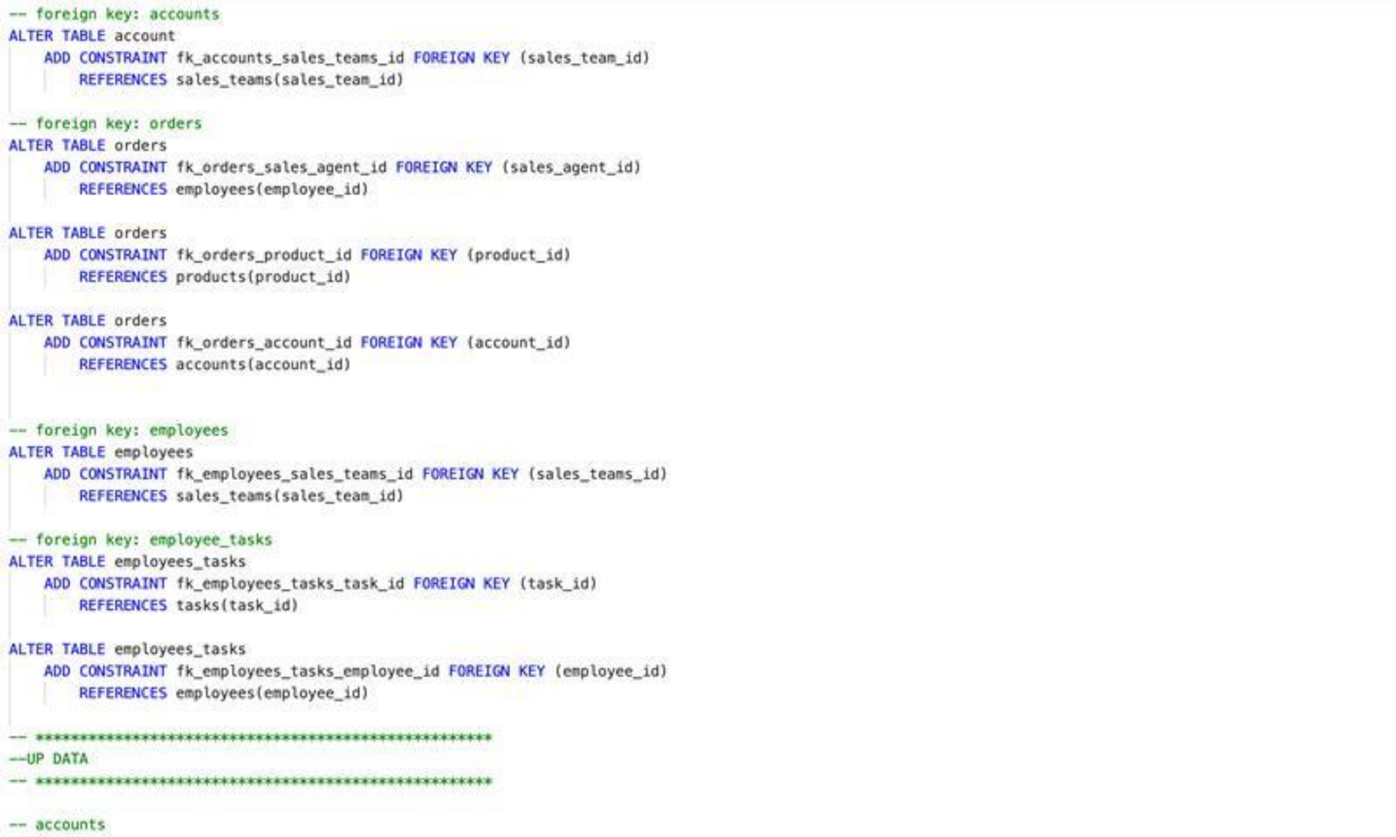
```
-- ******************************************************
--DOWN
-- ******************************************************

DROP TABLE IF EXISTS accounts

DROP TABLE IF EXISTS orders

DROP TABLE IF EXISTS products

DROP TABLE IF EXISTS employees

DROP TABLE IF EXISTS tasks

DROP TABLE IF EXISTS sales_teams

DROP TABLE IF EXISTS employees_tasks

-- foreign key: accounts
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS
    WHERE CONSTRAINT_NAME='fk_accounts_sales_teams_id')
    ALTER TABLE accounts DROP CONSTRAINT fk_accounts_sales_teams_id

-- foreign key: orders
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS
    WHERE CONSTRAINT_NAME='fk_orders_sales_agent_id')
    ALTER TABLE orders DROP CONSTRAINT fk_orders_sales_agent_id

IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS
    WHERE CONSTRAINT_NAME='fk_orders_product_id')
    ALTER TABLE orders DROP CONSTRAINT fk_orders_product_id

IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS
    WHERE CONSTRAINT_NAME='fk_orders_account_id')
    ALTER TABLE orders DROP CONSTRAINT fk_orders_account_id
```



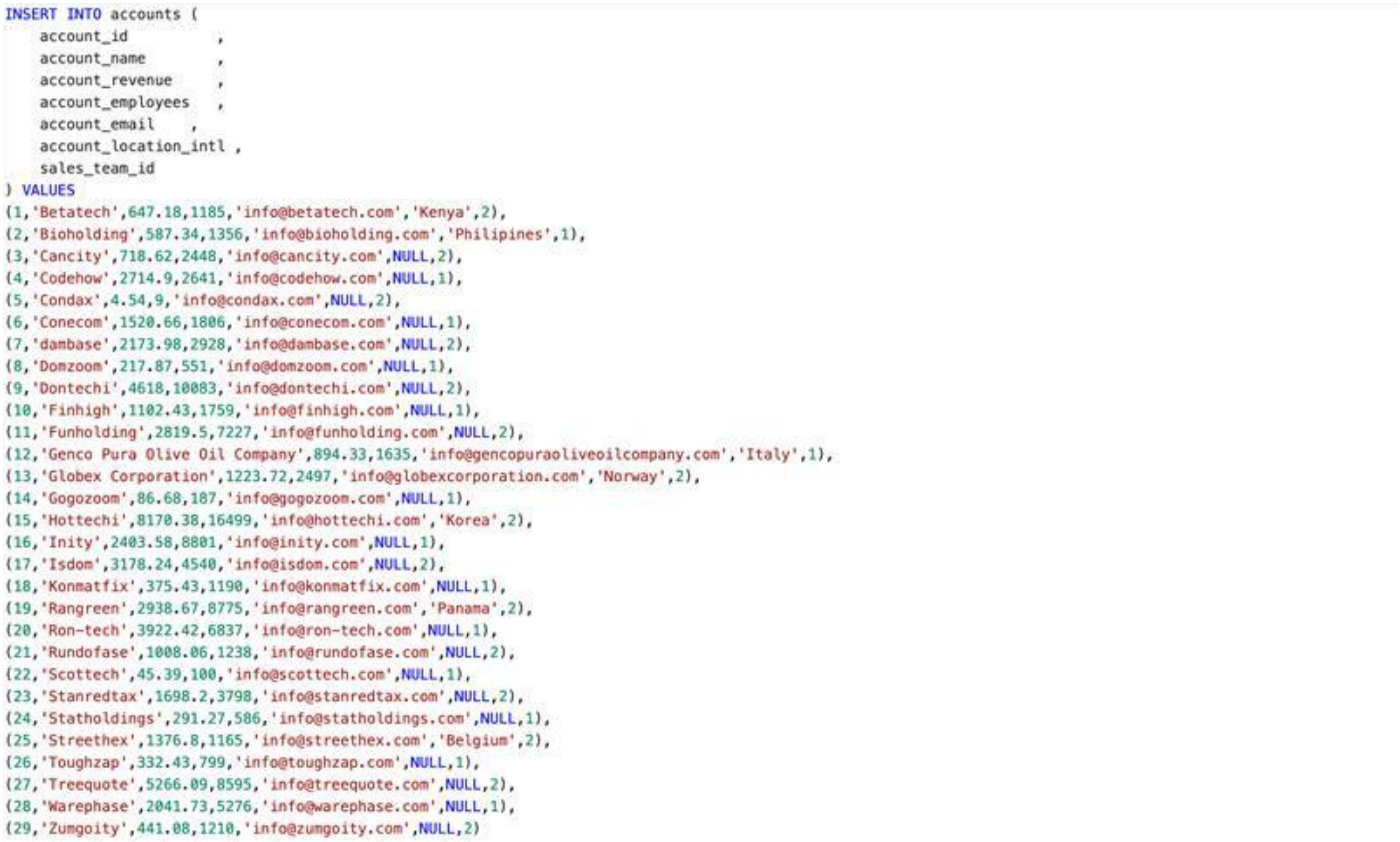
```
-- foreign key: employees
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS
    WHERE CONSTRAINT_NAME='fk_employees_sales_teams_id')
    ALTER TABLE employees DROP CONSTRAINT fk_employees_sales_teams_id

-- foreign key: employee_tasks
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS
    WHERE CONSTRAINT_NAME='fk_employees_tasks_task_id')
    ALTER TABLE employees_tasks DROP CONSTRAINT fk_employees_tasks_task_id

IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS
    WHERE CONSTRAINT_NAME='fk_employees_tasks_employee_id')
    ALTER TABLE employees_tasks DROP CONSTRAINT fk_employees_tasks_employee_id

-- ******************************************************
-- UP META
-- ******************************************************

CREATE TABLE accounts (
    account_id          INT IDENTITY NOT NULL,
    account_name        VARCHAR(50),
    account_revenue     DECIMAL,
    account_employees   INT,
    account_email       VARCHAR(40),
    account_location_intl VARCHAR(20),
    sales_team_id       INT,
    CONSTRAINT pk_accounts_account_id PRIMARY KEY(account_id),
    CONSTRAINT u_accounts_account_name UNIQUE(account_name),
    CONSTRAINT u_accounts_account_email UNIQUE(account_email)
)

CREATE TABLE orders (
    order_id            INT IDENTITY NOT NULL,
    product_id          INT NOT NULL,
    account_id          INT NOT NULL,
    sales_agent_id      INT NOT NULL,
    order_value         DECIMAL,
    order_date          DATETIME,
    CONSTRAINT pk_orders_order_id PRIMARY KEY(order_id)
)
```



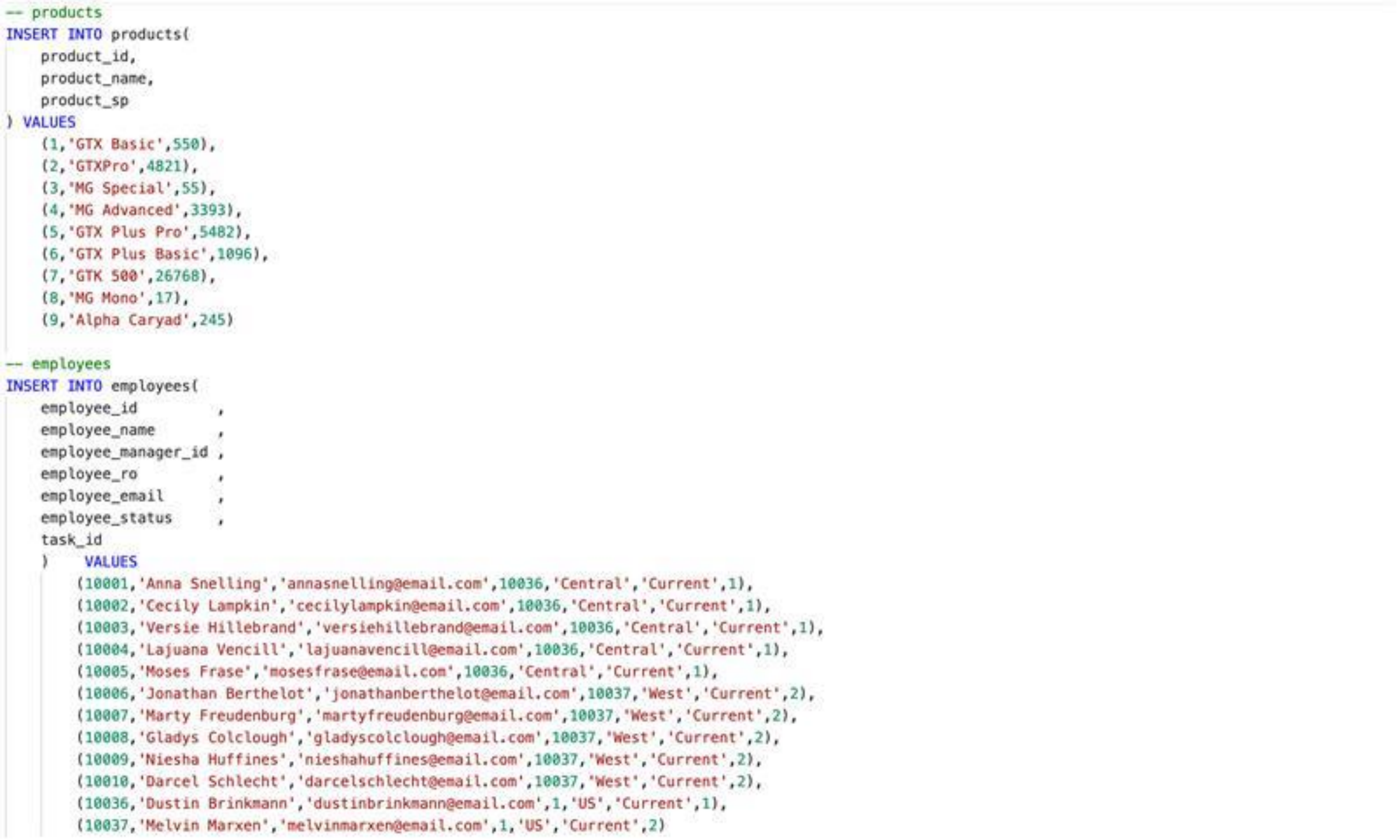
```
CREATE TABLE products(
    product_id          INT IDENTITY(1,1) NOT NULL,
    product_name        VARCHAR(50) NOT NULL,
    product_sp          INT NOT NULL,
    CONSTRAINT pk_products_product_id   PRIMARY KEY(product_id),
    CONSTRAINT u_products_product_name  UNIQUE(product_name)
)

CREATE TABLE employees(
    employee_id         INT IDENTITY NOT NULL,
    employee_name       VARCHAR(50) NOT NULL,
    employee_manager_id INT NOT NULL,
    employee_ro         VARCHAR(20) NOT NULL,
    employee_status     VARCHAR(50) NOT NULL,
    employee_email      VARCHAR(50) NULL,
    task_id             INT NULL,
    CONSTRAINT pk_employees_employee_id PRIMARY KEY(employee_id)
)

CREATE TABLE tasks(
    task_id             INT IDENTITY(1,1) NOT NULL,
    task_desc           VARCHAR(50) NOT NULL,
    assigned_to_team    INT NOT NULL,
    completed           CHAR(1) NOT NULL,
    CONSTRAINT pk_tasks_task_id PRIMARY KEY(task_id),
    CONSTRAINT ck_task_completed CHECK(completed IN ('Y', 'N'))
)

CREATE TABLE sales_teams (
    sales_team_id           INT IDENTITY NOT NULL,
    sales_team_manager_id   INT NOT NULL,
    CONSTRAINT pk_sales_teams_sales_team_id PRIMARY KEY(sales_team_id)
)

CREATE TABLE employees_tasks (
    employee_id         INT NOT NULL,
    task_id             INT NOT NULL,
    CONSTRAINT pk_employee_tasks PRIMARY KEY(employee_id, task_id)
)
```



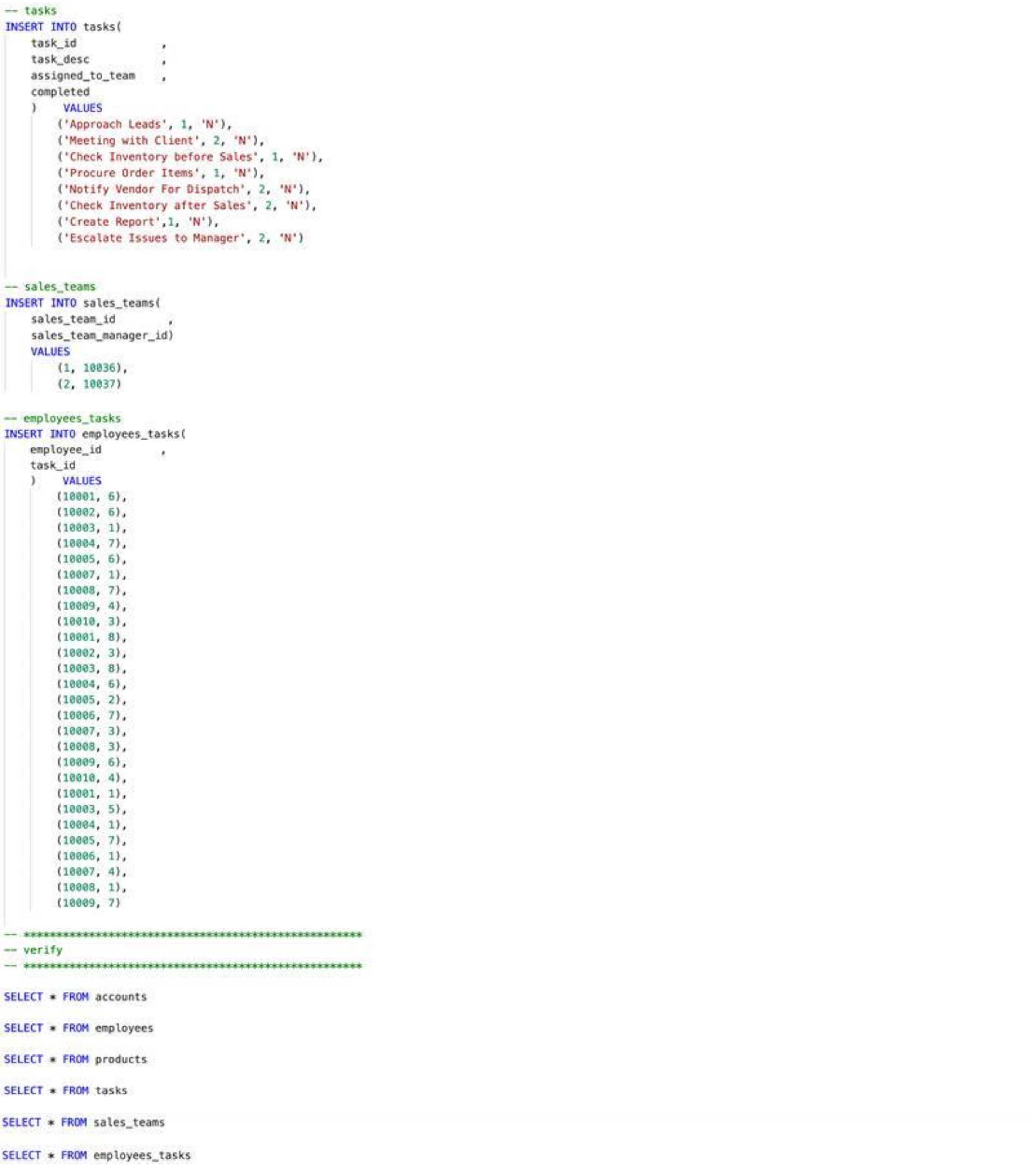
```
-- foreign key: accounts
ALTER TABLE account
    ADD CONSTRAINT fk_accounts_sales_teams_id FOREIGN KEY (sales_team_id)
        REFERENCES sales_teams(sales_team_id)

-- foreign key: orders
ALTER TABLE orders
    ADD CONSTRAINT fk_orders_sales_agent_id FOREIGN KEY (sales_agent_id)
        REFERENCES employees(employee_id)

ALTER TABLE orders
    ADD CONSTRAINT fk_orders_product_id FOREIGN KEY (product_id)
        REFERENCES products(product_id)

ALTER TABLE orders
    ADD CONSTRAINT fk_orders_account_id FOREIGN KEY (account_id)
        REFERENCES accounts(account_id)


-- foreign key: employees
ALTER TABLE employees
    ADD CONSTRAINT fk_employees_sales_teams_id FOREIGN KEY (sales_teams_id)
        REFERENCES sales_teams(sales_team_id)

-- foreign key: employee_tasks
ALTER TABLE employees_tasks
    ADD CONSTRAINT fk_employees_tasks_task_id FOREIGN KEY (task_id)
        REFERENCES tasks(task_id)

ALTER TABLE employees_tasks
    ADD CONSTRAINT fk_employees_tasks_employee_id FOREIGN KEY (employee_id)
        REFERENCES employees(employee_id)

-- ****************************************************
--UP DATA
-- ****************************************************

-- accounts
```



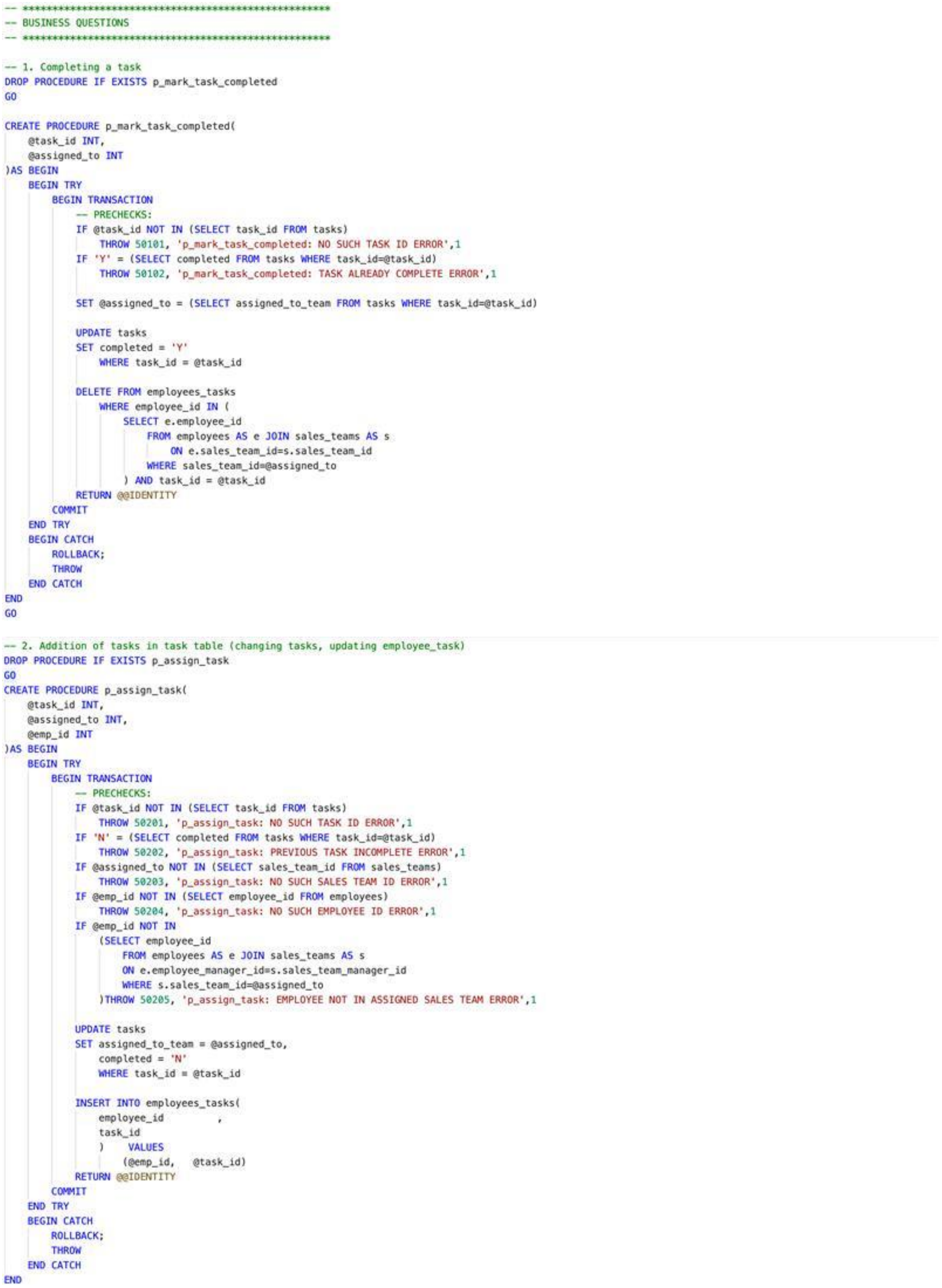
```sql
INSERT INTO accounts (
    account_id          ,
    account_name        ,
    account_revenue     ,
    account_employees   ,
    account_email     ,
    account_location_intl ,
    sales_team_id
) VALUES
(1,'Betatech',647.18,1185,'info@betatech.com','Kenya',2),
(2,'Bioholding',587.34,1356,'info@bioholding.com','Philipines',1),
(3,'Cancity',718.62,2448,'info@cancity.com',NULL,2),
(4,'Codehow',2714.9,2641,'info@codehow.com',NULL,1),
(5,'Condax',4.54,9,'info@condax.com',NULL,2),
(6,'Conecom',1520.66,1806,'info@conecom.com',NULL,1),
(7,'dambase',2173.98,2928,'info@dambase.com',NULL,2),
(8,'Domzoom',217.87,551,'info@domzoom.com',NULL,1),
(9,'Dontechi',4618,10083,'info@dontechi.com',NULL,2),
(10,'Finhigh',1102.43,1759,'info@finhigh.com',NULL,1),
(11,'Funholding',2819.5,7227,'info@funholding.com',NULL,2),
(12,'Genco Pura Olive Oil Company',894.33,1635,'info@gencopuraoliveoilcompany.com','Italy',1),
(13,'Globex Corporation',1223.72,2497,'info@globexcorporation.com','Norway',2),
(14,'Gogozoom',86.68,187,'info@gogozoom.com',NULL,1),
(15,'Hottechi',8170.38,16499,'info@hottechi.com','Korea',2),
(16,'Inity',2403.58,8801,'info@inity.com',NULL,1),
(17,'Isdom',3178.24,4540,'info@isdom.com',NULL,2),
(18,'Konmatfix',375.43,1190,'info@konmatfix.com',NULL,1),
(19,'Rangreen',2938.67,8775,'info@rangreen.com','Panama',2),
(20,'Ron-tech',3922.42,6837,'info@ron-tech.com',NULL,1),
(21,'Rundofase',1008.06,1238,'info@rundofase.com',NULL,2),
(22,'Scottech',45.39,100,'info@scottech.com',NULL,1),
(23,'Stanredtax',1698.2,3798,'info@stanredtax.com',NULL,2),
(24,'Statholdings',291.27,586,'info@statholdings.com',NULL,1),
(25,'Streethex',1376.8,1165,'info@streethex.com','Belgium',2),
(26,'Toughzap',332.43,799,'info@toughzap.com',NULL,1),
(27,'Treequote',5266.09,8595,'info@treequote.com',NULL,2),
(28,'Warephase',2041.73,5276,'info@warephase.com',NULL,1),
(29,'Zumgoity',441.08,1210,'info@zumgoity.com',NULL,2)
```



```sql
-- products
INSERT INTO products(
    product_id,
    product_name,
    product_sp
) VALUES
    (1,'GTX Basic',550),
    (2,'GTXPro',4821),
    (3,'MG Special',55),
    (4,'MG Advanced',3393),
    (5,'GTX Plus Pro',5482),
    (6,'GTX Plus Basic',1096),
    (7,'GTK 500',26768),
    (8,'MG Mono',17),
    (9,'Alpha Caryad',245)

-- employees
INSERT INTO employees(
    employee_id          ,
    employee_name        ,
    employee_manager_id ,
    employee_ro          ,
    employee_email       ,
    employee_status      ,
    task_id
    )    VALUES
        (10001,'Anna Snelling','annasnelling@email.com',10036,'Central','Current',1),
        (10002,'Cecily Lampkin','cecilylampkin@email.com',10036,'Central','Current',1),
        (10003,'Versie Hillebrand','versiehillebrand@email.com',10036,'Central','Current',1),
        (10004,'Lajuana Vencill','lajuanavencill@email.com',10036,'Central','Current',1),
        (10005,'Moses Frase','mosesfrase@email.com',10036,'Central','Current',1),
        (10006,'Jonathan Berthelot','jonathanberthelot@email.com',10037,'West','Current',2),
        (10007,'Marty Freudenburg','martyfreudenburg@email.com',10037,'West','Current',2),
        (10008,'Gladys Colclough','gladyscolclough@email.com',10037,'West','Current',2),
        (10009,'Niesha Huffines','nieshahuffines@email.com',10037,'West','Current',2),
        (10010,'Darcel Schlecht','darcelschlecht@email.com',10037,'West','Current',2),
        (10036,'Dustin Brinkmann','dustinbrinkmann@email.com',1,'US','Current',1),
        (10037,'Melvin Marxen','melvinmarxen@email.com',1,'US','Current',2)
```

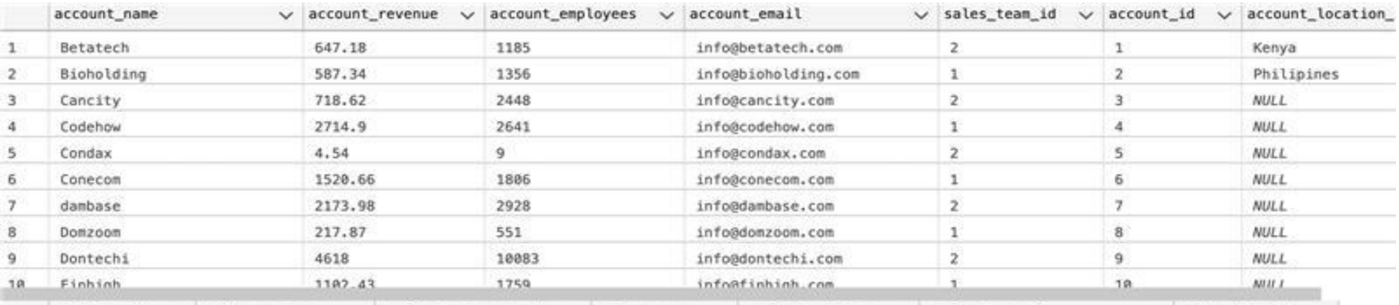


```
-- tasks
INSERT INTO tasks(
    task_id             ,
    task_desc           ,
    assigned_to_team    ,
    completed
    )    VALUES
        ('Approach Leads', 1, 'N'),
        ('Meeting with Client', 2, 'N'),
        ('Check Inventory before Sales', 1, 'N'),
        ('Procure Order Items', 1, 'N'),
        ('Notify Vendor For Dispatch', 2, 'N'),
        ('Check Inventory after Sales', 2, 'N'),
        ('Create Report',1, 'N'),
        ('Escalate Issues to Manager', 2, 'N')


-- sales_teams
INSERT INTO sales_teams(
    sales_team_id          ,
    sales_team_manager_id)
    VALUES
        (1, 10036),
        (2, 10037)

-- employees_tasks
INSERT INTO employees_tasks(
    employee_id          ,
    task_id
    )    VALUES
        (10001, 6),
        (10002, 6),
        (10003, 1),
        (10004, 7),
        (10005, 6),
        (10007, 1),
        (10008, 7),
        (10009, 4),
        (10010, 3),
        (10001, 8),
        (10002, 3),
        (10003, 8),
        (10004, 6),
        (10005, 2),
        (10006, 7),
        (10007, 3),
        (10008, 3),
        (10009, 6),
        (10010, 4),
        (10001, 1),
        (10003, 5),
        (10004, 1),
        (10005, 7),
        (10006, 1),
        (10007, 4),
        (10008, 1),
        (10009, 7)

-- ******************************************************
-- verify
-- ******************************************************

SELECT * FROM accounts

SELECT * FROM employees

SELECT * FROM products

SELECT * FROM tasks

SELECT * FROM sales_teams

SELECT * FROM employees_tasks
```

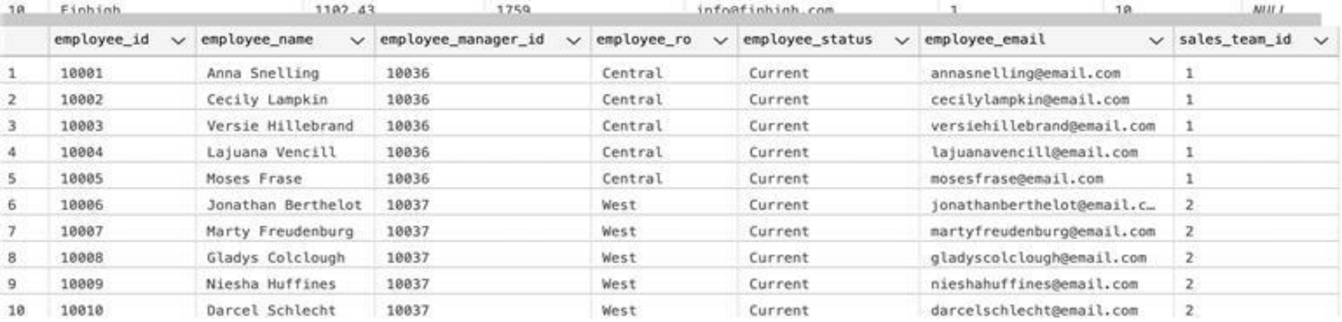


```sql
-- ****************************************************
-- BUSINESS QUESTIONS
-- ****************************************************

-- 1. Completing a task
DROP PROCEDURE IF EXISTS p_mark_task_completed
GO

CREATE PROCEDURE p_mark_task_completed(
    @task_id INT,
    @assigned_to INT
)AS BEGIN
    BEGIN TRY
        BEGIN TRANSACTION
            -- PRECHECKS:
            IF @task_id NOT IN (SELECT task_id FROM tasks)
                THROW 50101, 'p_mark_task_completed: NO SUCH TASK ID ERROR',1
            IF 'Y' = (SELECT completed FROM tasks WHERE task_id=@task_id)
                THROW 50102, 'p_mark_task_completed: TASK ALREADY COMPLETE ERROR',1

            SET @assigned_to = (SELECT assigned_to_team FROM tasks WHERE task_id=@task_id)

            UPDATE tasks
            SET completed = 'Y'
                WHERE task_id = @task_id

            DELETE FROM employees_tasks
                WHERE employee_id IN (
                    SELECT e.employee_id
                        FROM employees AS e JOIN sales_teams AS s
                            ON e.sales_team_id=s.sales_team_id
                        WHERE sales_team_id=@assigned_to
                    ) AND task_id = @task_id
            RETURN @@IDENTITY
        COMMIT
    END TRY
    BEGIN CATCH
        ROLLBACK;
        THROW
    END CATCH
END
GO

-- 2. Addition of tasks in task table (changing tasks, updating employee_task)
DROP PROCEDURE IF EXISTS p_assign_task
GO
CREATE PROCEDURE p_assign_task(
    @task_id INT,
    @assigned_to INT,
    @emp_id INT
)AS BEGIN
    BEGIN TRY
        BEGIN TRANSACTION
            -- PRECHECKS:
            IF @task_id NOT IN (SELECT task_id FROM tasks)
                THROW 50201, 'p_assign_task: NO SUCH TASK ID ERROR',1
            IF 'N' = (SELECT completed FROM tasks WHERE task_id=@task_id)
                THROW 50202, 'p_assign_task: PREVIOUS TASK INCOMPLETE ERROR',1
            IF @assigned_to NOT IN (SELECT sales_team_id FROM sales_teams)
                THROW 50203, 'p_assign_task: NO SUCH SALES TEAM ID ERROR',1
            IF @emp_id NOT IN (SELECT employee_id FROM employees)
                THROW 50204, 'p_assign_task: NO SUCH EMPLOYEE ID ERROR',1
            IF @emp_id NOT IN
                (SELECT employee_id
                    FROM employees AS e JOIN sales_teams AS s
                    ON e.employee_manager_id=s.sales_team_manager_id
                    WHERE s.sales_team_id=@assigned_to
                )THROW 50205, 'p_assign_task: EMPLOYEE NOT IN ASSIGNED SALES TEAM ERROR',1

            UPDATE tasks
            SET assigned_to_team = @assigned_to,
                completed = 'N'
                WHERE task_id = @task_id

            INSERT INTO employees_tasks(
                employee_id         ,
                task_id
                )    VALUES
                    (@emp_id,   @task_id)
            RETURN @@IDENTITY
        COMMIT
    END TRY
    BEGIN CATCH
        ROLLBACK;
        THROW
    END CATCH
END
```

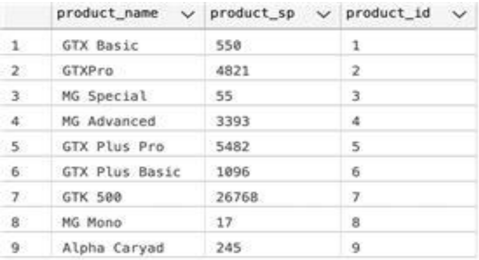
# 9) Screenshot of Tables



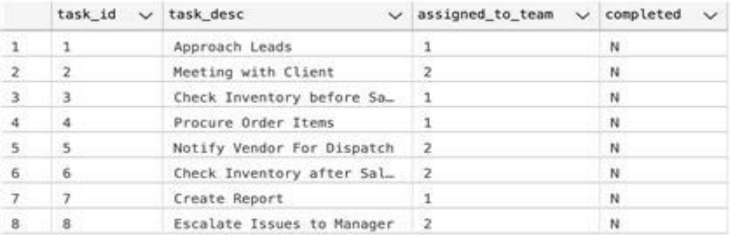
| | account_name | account_revenue | account_employees | account_email | sales_team_id | account_id | account_location_ |
|---|---|---|---|---|---|---|---|
| 1 | Betatech | 647.18 | 1185 | info@betatech.com | 2 | 1 | Kenya |
| 2 | Bioholding | 587.34 | 1356 | info@bioholding.com | 1 | 2 | Philipines |
| 3 | Cancity | 718.62 | 2448 | info@cancity.com | 2 | 3 | NULL |
| 4 | Codehow | 2714.9 | 2641 | info@codehow.com | 1 | 4 | NULL |
| 5 | Condax | 4.54 | 9 | info@condax.com | 2 | 5 | NULL |
| 6 | Conecom | 1520.66 | 1806 | info@conecom.com | 1 | 6 | NULL |
| 7 | dambase | 2173.98 | 2928 | info@dambase.com | 2 | 7 | NULL |
| 8 | Domzoom | 217.87 | 551 | info@domzoom.com | 1 | 8 | NULL |
| 9 | Dontechi | 4618 | 10083 | info@dontechi.com | 2 | 9 | NULL |
| 10 | Finhigh | 1102.43 | 1759 | info@finhigh.com | 1 | 10 | NULL |



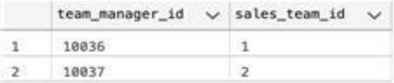
| | employee_id | employee_name | employee_manager_id | employee_ro | employee_status | employee_email | sales_team_id |
|---|---|---|---|---|---|---|---|
| 1 | 10001 | Anna Snelling | 10036 | Central | Current | annasnelling@email.com | 1 |
| 2 | 10002 | Cecily Lampkin | 10036 | Central | Current | cecilylampkin@email.com | 1 |
| 3 | 10003 | Versie Hillebrand | 10036 | Central | Current | versiehillebrand@email.com | 1 |
| 4 | 10004 | Lajuana Vencill | 10036 | Central | Current | lajuanavencill@email.com | 1 |
| 5 | 10005 | Moses Frase | 10036 | Central | Current | mosesfrase@email.com | 1 |
| 6 | 10006 | Jonathan Berthelot | 10037 | West | Current | jonathanberthelot@email.c… | 2 |
| 7 | 10007 | Marty Freudenburg | 10037 | West | Current | martyfreudenburg@email.com | 2 |
| 8 | 10008 | Gladys Colclough | 10037 | West | Current | gladyscolclough@email.com | 2 |
| 9 | 10009 | Niesha Huffines | 10037 | West | Current | nieshahuffines@email.com | 2 |
| 10 | 10010 | Darcel Schlecht | 10037 | West | Current | darcelschlecht@email.com | 2 |



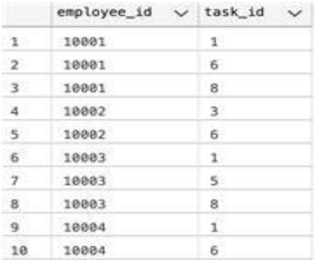
| | product_name | product_sp | product_id |
|---|---|---|---|
| 1 | GTX Basic | 550 | 1 |
| 2 | GTXPro | 4821 | 2 |
| 3 | MG Special | 55 | 3 |
| 4 | MG Advanced | 3393 | 4 |
| 5 | GTX Plus Pro | 5482 | 5 |
| 6 | GTX Plus Basic | 1096 | 6 |
| 7 | GTK 500 | 26768 | 7 |
| 8 | MG Mono | 17 | 8 |
| 9 | Alpha Caryad | 245 | 9 |



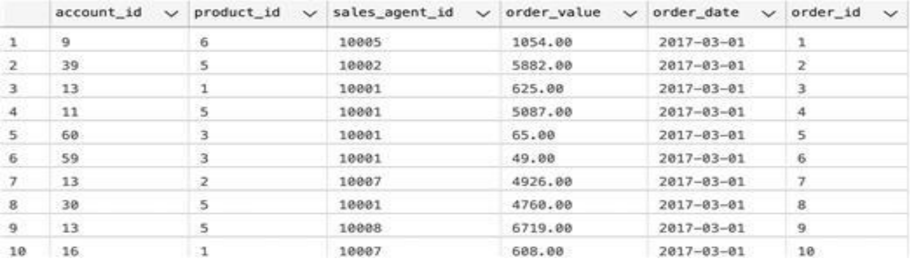
| | task_id | task_desc | assigned_to_team | completed |
|---|---|---|---|---|
| 1 | 1 | Approach Leads | 1 | N |
| 2 | 2 | Meeting with Client | 2 | N |
| 3 | 3 | Check Inventory before Sa… | 1 | N |
| 4 | 4 | Procure Order Items | 1 | N |
| 5 | 5 | Notify Vendor For Dispatch | 2 | N |
| 6 | 6 | Check Inventory after Sal… | 2 | N |
| 7 | 7 | Create Report | 1 | N |
| 8 | 8 | Escalate Issues to Manager | 2 | N |



| | team_manager_id | sales_team_id |
|---|---|---|
| 1 | 10036 | 1 |
| 2 | 10037 | 2 |



| | employee_id | task_id |
|---|---|---|
| 1 | 10001 | 1 |
| 2 | 10001 | 6 |
| 3 | 10001 | 8 |
| 4 | 10002 | 3 |
| 5 | 10002 | 6 |
| 6 | 10003 | 1 |
| 7 | 10003 | 5 |
| 8 | 10003 | 8 |
| 9 | 10004 | 1 |
| 10 | 10004 | 6 |



| | account_id | product_id | sales_agent_id | order_value | order_date | order_id |
|---|---|---|---|---|---|---|
| 1 | 9 | 6 | 10005 | 1054.00 | 2017-03-01 | 1 |
| 2 | 39 | 5 | 10002 | 5882.00 | 2017-03-01 | 2 |
| 3 | 13 | 1 | 10001 | 625.00 | 2017-03-01 | 3 |
| 4 | 11 | 5 | 10001 | 5087.00 | 2017-03-01 | 4 |
| 5 | 60 | 3 | 10001 | 65.00 | 2017-03-01 | 5 |
| 6 | 59 | 3 | 10001 | 49.00 | 2017-03-01 | 6 |
| 7 | 13 | 2 | 10007 | 4926.00 | 2017-03-01 | 7 |
| 8 | 30 | 5 | 10001 | 4760.00 | 2017-03-01 | 8 |
| 9 | 13 | 5 | 10008 | 6719.00 | 2017-03-01 | 9 |
| 10 | 16 | 1 | 10007 | 608.00 | 2017-03-01 | 10 |

## 10) Team Log

| Team Member | Work Done |
|---|---|
| Manan Vora | Collaborated with Aditya to make the logical model, Created Tables, Created Application Screens on Adobe XD |
| Aditya Shah | Coordinated with the team to complete the project promptly,<br>Collaborated with Manan to make the logical model.<br>Wrote the script for Up/Down for queries and inserting data into the database. |
| Amesh Gharat | Made the conceptual model,<br>Wrote the documentation for the project and made PPT. |