CREATE DATABASE "POSSystem"

WITH

OWNER = postgres

ENCODING = 'UTF8'

CONNECTION LIMIT = -1;

- **Create tables**
    - CREATE TABLE employee(

employee_number INT NOT NULL,

employee_firstname varchar(30) NOT NULL,

employee_lastname varchar(30) NOT NULL,

department_name varchar(40) NOT NULL,

more_reward_number INT NOT NULL,

job_title varchar(40) NOT NULL,

hiring_date date NOT NULL,

PRIMARY KEY(employee_number)

);

    - CREATE TABLE supplier(

supplier_register_number varchar(20) NOT NULL,

supplier_name varchar(20) NOT NULL,

supplier_address varchar(40) NOT NULL,

supplier_postalcode varchar(10) NOT NULL,

supplier_city varchar(20) NOT NULL,

PRIMARY KEY(supplier_register_number)

);

    - CREATE TABLE department(

department_name varchar(30) NOT NULL,

department_phone_num varchar(20) NOT NULL,

department_location varchar(30) NOT NULL,

PRIMARY KEY(department_name)

);

    - CREATE TABLE cash_register(

cash_register_number varchar(30) NOT NULL,

employee_number INT NOT NULL,

last_signed_in_date timestamp NOT NULL,

last_signed_out_date timestamp NOT NULL,

PRIMARY KEY(cash_register_number)

);

- CREATE TABLE receipt(

receipt_number varchar(30) NOT NULL,

receipt_date timestamp NOT NULL,

cash_register_number varchar(30) NOT NULL,

more_reward_number INT,

total_amount decimal(12,2) NOT NULL,

PRIMARY KEY(receipt_number)

);

- CREATE TABLE item(

item_code varchar(20) NOT NULL,

item_name varchar(30) NOT NULL,

supplier_name varchar(30) NOT NULL,

receipt_number varchar(30) NOT NULL,

amount decimal(12, 2) NOT NULL,

offer_code varchar(30),

PRIMARY KEY(item_code)

);

- CREATE TABLE customer_more_reward(

more_reward_number INT NOT NULL,

opening_date timestamp NOT NULL,

number_of_points INT NOT NULL,

last_used_date timestamp NOT NULL,

cardholder_firstname varchar(20) NOT NULL,

cardholder_lastname varchar(20) NOT NULL,

cardholder_email varchar(30),

cardholder_address varchar(40),

cardholder_postalcode varchar(10),

cardholder_city varchar(20),

cardholder_phonenumber INT,

PRIMARY KEY(more_reward_number)

);

- CREATE TABLE offer(

offer_code varchar(20) NOT NULL,

offer_beginning_date timestamp NOT NULL,

offer_ending_date timestamp NOT NULL,

offer_amount decimal(12,2) NOT NULL,

points_required INT,

PRIMARY KEY(offer_code)

);

- **Modify tables**
  - ALTER TABLE employee

ADD CONSTRAINT FK_department

FOREIGN KEY (department_name) REFERENCES department(department_name);

- ALTER TABLE department

ALTER COLUMN department_phone_num TYPE varchar(20);

- ALTER TABLE employee

ADD CONSTRAINT FK_customer

FOREIGN KEY (more_reward_number) REFERENCES customer_more_reward(more_reward_number);

- *To change data type for a table*
  - ALTER TABLE employee

DROP CONSTRAINT FK_customer;

- ALTER TABLE employee

ALTER COLUMN more_reward_number TYPE varchar(20);

- ALTER TABLE customer_more_reward

ALTER COLUMN more_reward_number TYPE varchar(20);

- ALTER TABLE customer_more_reward

ALTER COLUMN number_of_points TYPE numeric;

- ALTER TABLE customer_more_reward

ALTER COLUMN cardholder_phonenumber TYPE varchar(15);

- INSERT INTO customer_more_reward(more_reward_number, opening_date, number_of_points, last_used_date, cardholder_firstname, cardholder_lastname, cardholder_email, cardholder_address, cardholder_postalcode, cardholder_city, cardholder_phonenumber)

VALUES

('4808008000', 'July-20-2014', 9000, 'October-13-2018', 'Liliana', 'Tang', 'ltang@chandos.com', '9604-20 Avenue', 'T6H4Y7', 'Edmonton', '7806801219');

- INSERT INTO department (department_name, department_phone_num, department_location)

VALUES

('front end', '7804267069', '9th St Jasper Avenue');

- INSERT INTO customer_more_reward(more_reward_number, opening_date, number_of_points, last_used_date, cardholder_firstname, cardholder_lastname, cardholder_email, cardholder_address, cardholder_postalcode, cardholder_city, cardholder_phonenumber)

VALUES

('4808008001', 'April-2-2004', 90000, 'October-14-2018', 'Chris', 'Smith', 'chris.smith@hotmail.com', '5045 Whitemud Drive', 'T6M4Y3', 'Edmonton', '5876801219');

- INSERT INTO department (department_name, department_phone_num, department_location)

VALUES

('meat', '7804267068', '9th St Jasper Avenue');

- INSERT INTO employee(employee_number, employee_firstname, employee_lastname,

department_name, more_reward_number, job_title, hiring_date)

VALUES

(1, 'Liliana', 'Tang', 'front end', '4808008000', 'front end specialist', 'July-20-2014');

- INSERT INTO customer_more_reward(more_reward_number, opening_date, number_of_points, last_used_date, cardholder_firstname, cardholder_lastname, cardholder_email, cardholder_address, cardholder_postalcode, cardholder_city, cardholder_phonenumber)

VALUES

('4808009000', 'July-29-2004', 8000000, 'October-14-2018', 'Ricky', 'Elizabeth', 'ricky.elizabeth@hotmail.com', '5860 166 Avenue', 'T6I4Y9', 'Edmonton', '5876901219');

- INSERT INTO employee(employee_number, employee_firstname, employee_lastname, department_name, more_reward_number, job_title, hiring_date)

VALUES

(2, 'Ricky', 'Elizabeth', 'front end', '4808009000', 'front end specialist', 'July-29-2014');

- INSERT INTO employee(employee_number, employee_firstname, employee_lastname, department_name, more_reward_number, job_title, hiring_date)

VALUES

(3, 'Richard', 'Elizabeth', 'meat', '4808009010', 'meat specialist', 'July-29-2014');

- INSERT INTO customer_more_reward(more_reward_number, opening_date, number_of_points, last_used_date, cardholder_firstname, cardholder_lastname, cardholder_email, cardholder_address, cardholder_postalcode, cardholder_city, cardholder_phonenumber)

VALUES

('4808009003', 'August-20-2018', 9000,'August-20-2018' , 'Victoria', 'Tran', 'victoria.tran@gmail.com', '9904-30 Avenue', 'T5H4Y9', 'Edmonton', '5876911219');

- INSERT INTO customer_more_reward(more_reward_number, opening_date, number_of_points, last_used_date, cardholder_firstname, cardholder_lastname, cardholder_email, cardholder_address, cardholder_postalcode, cardholder_city, cardholder_phonenumber)

VALUES

('4808009002', 'August-20-2014', 9000,'August-20-2014' , 'Ryan', 'Tran', 'ryan.tran@gmail.com', '9604-30 Avenue', 'T6H4Y9', 'Edmonton', '7806911219');

- INSERT INTO cash_register(cash_register_number, employee_number, last_signed_in_date, last_signed_out_date)

VALUES

(1, 2, 'October 13, 2018', 'October 13, 2018');

INSERT INTO receipt(receipt_number, receipt_date, cash_register_number, more_reward_number, total_amount)

VALUES ('AB9100000', 'October 13, 2018', 1, '4808008000', 125.56);

- INSERT INTO customer_more_reward(more_reward_number, opening_date, number_of_points, last_used_date, cardholder_firstname, cardholder_lastname, cardholder_email, cardholder_address, cardholder_postalcode, cardholder_city, cardholder_phonenumber)

VALUES

('4808009010', 'July-29-2014', 3000, 'October-14-2018', 'Richard', 'Elizabeth', 'richard.elizabeth@hotmail.com', '5960 166 Avenue', 'T9I4Y9', 'Edmonton', '5876901219');

- **Add foreign keys**
  - ALTER TABLE item

ADD CONSTRAINT FK_offer

FOREIGN KEY (offer_code) REFERENCES offer(offer_code);

- ALTER TABLE item

ADD CONSTRAINT FK_receipt

FOREIGN KEY (receipt_number) REFERENCES receipt(receipt_number);

- ALTER TABLE cash_register

ADD CONSTRAINT FK_employee

FOREIGN KEY (employee_number) REFERENCES employee(employee_number);

- UPDATE employee

SET hourly_rate = 15.20 WHERE employee_number = 1;

- UPDATE employee

SET hourly_rate = 15.00 WHERE employee_number = 2;

- UPDATE employee

SET working_hours = 10000 WHERE employee_number = 1;

- **Delete a table** – I then realized POS System doesn't need a supplier table
  - DROP TABLE supplier;

- **Simple SELECTs on single tables**
  - Find out the last name of an employee who has first name as Liliana
    - SELECT employee_lastname FROM employee WHERE employee.employee_firstname = 'Liliana';

	employee_lastname character varying (30)
1	Tang

- **Select a subset of the fields in a table**
  - Display all employees' more reward card number
    - SELECT more_reward_number FROM employee

	more_reward_number character varying (20)
1	4808008000
2	4808009000

- **Join at least two tables together using a WHERE clause, an ON clause or some form of JOIN clause (e.g. INNER JOIN, OUTER JOIN, NATURAL JOIN)**
  - Find the total amount Liliana have spent in the store
    - SELECT receipt.total_amount FROM employee INNER JOIN receipt

      ON receipt.more_reward_number = employee.more_reward_number

      WHERE employee.employee_firstname = 'Liliana'

total_amount numeric (12,2)
125.56

- **Sort results, ascending and descending**
  - Display all employees sorting by their hiring date, from the longest-serving employees
    - SELECT * FROM employee

      ORDER BY hiring_date

employee_number integer	employee_firstname character varying (30)	employee_lastname character varying (30)	department_name character varying (40)	more_reward_number character varying (20)	job_title character varying (40)	hiring_da date
1	Liliana	Tang	front end	4808008000	front end specialist	2014-07-2
2	Ricky	Elizabeth	front end	4808009000	front end specialist	2014-07-2

    - SELECT * FROM employee
      order by hiring_date DESC

	employee_number integer	employee_firstname character varying (30)	employee_lastname character varying (30)	department_name character varying (40)	more_reward_number character varying (20)	job_title character varying (40)	h d
1	2	Ricky	Elizabeth	front end	4808009000	front end specialist	2(
2	1	Liliana	Tang	front end	4808008000	front end specialist	2(

- **Use functions**
  - When an employee works on national holidays, they are supposed to get double paid.
    - CREATE FUNCTION double_pay(employee) RETURNS money AS $$

      SELECT $1.hourly_rate * 2 AS hourly_rate;

      $$ LANGUAGE SQL;

SELECT employee_number, double_pay(employee.*) AS holidays

FROM employee

- **Use a GROUP clause to group data & Use a calculation to create a named generated field using AS (e.g., SELECT COUNT(*) AS myCount)**
  - Count the total number of employees per department in the store
    - SELECT department_name, COUNT(*) AS num_of_employees

      FROM employee

      GROUP BY department_name;

department_name character varying (40)	num_of_employees bigint
meat	1
front end	2

- **Use COUNT, AVERAGE or other group-oriented commands in grouped data**
  - Calculate the average of points all regular customers have earned (regular customers = customers who own a more reward card)
    - SELECT AVG(number_of_points) from customer_more_reward

	avg numeric
1	1623400.000000000000

- **A query to join at least three tables using commands such as WHERE, UNION, and JOIN**
  - Display all employees' information and their shopping information and their department contact info
    - SELECT employee.employee_number, CONCAT(employee.employee_firstname, ' ', employee.employee_lastname),employee.hiring_date,

      customer_more_reward.number_of_points, department.department_name, department.department_phone_num

FROM employee

INNER JOIN customer_more_reward ON employee.more_reward_number = customer_more_reward.more_reward_number

INNER JOIN department ON department.department_name = employee.department_name

employee_number integer	concat text	hiring_date date	number_of_points numeric	department_name character varying (30)	department_phone_nu character varying (20)
1	Liliana Tang	2014-07-20	9000	front end	7804267069
2	Ricky Elizabeth	2014-07-29	8000000	front end	7804267069
3	Richard Elizabeth	2014-07-29	3000	meat	7804267068

- **The same JOIN as above as a sub-query (e.g., SELECT x WHERE y IN (SELECT yy FROM**
    - Show all employees who once shopped in the store they work. Note that in this case, their hiring_date would be before the last_used_date in their More Reward card.
        - SELECT * FROM employee

        WHERE employee.hiring_date NOT IN

        (SELECT customer_more_reward.last_used_date FROM customer_more_reward)

employee_number integer	employee_firstname character varying (30)	employee_lastname character varying (30)	department_name character varying (40)	more_reward_number character varying (20)	job_title character varying (40)	hiring_date date	hourly_r money
1	Liliana	Tang	front end	4808008000	front end specialist	2014-07-20	[null]
2	Ricky	Elizabeth	front end	4808009000	front end specialist	2014-07-29	[null]

- **Re-format data using functions or casting (e.g., to provide a different data format, to convert the case of some text, or to format numbers as currency)**
    - Display all employees' full name working in the store

        - SELECT CONCAT(employee_firstname, ' ', employee_lastname) from employee

	concat text
1	Liliana Tang
2	Ricky Elizabeth