CREATE TABLE "User"(

"ID" SERIAL PRIMARY KEY,

"Username" VARCHAR(50) NOT NULL,

"Password" VARCHAR(50) NOT NULL,

"FirstName" VARCHAR(50) NOT NULL,

"MiddleName" VARCHAR(50),

"LastName" VARCHAR(50) NOT NULL,

"GivenName" VARCHAR(50),

"Addr1" VARCHAR(50) NOT NULL,

"Addr2" VARCHAR(50),

"Province" VARCHAR(50) NOT NULL,

"PostalCode" VARCHAR(10) NOT NULL,

"Sex" VARCHAR(3) NOT NULL,

"Phone" VARCHAR(15) NOT NULL,

"DOB" DATE NOT NULL,

"Email" VARCHAR(50) NOT NULL

);

CREATE TABLE "Table"(

"TableNo" SERIAL PRIMARY KEY,

"Location" VARCHAR(50) NOT NULL,

"isOccupied" BOOLEAN NOT NULL DEFAULT FALSE,

"waiter\_ID" INT,

FOREIGN KEY("waiter\_ID") REFERENCES "User"("ID")

ON DELETE SET NULL

ON UPDATE CASCADE

);

CREATE TABLE "Employee"(

"User\_ID" INT NOT NULL,

"Start\_Date" DATE NOT NULL,

"Job\_Title" VARCHAR(50) NOT NULL,

"Salary" NUMERIC(8, 2) NOT NULL,

"mgr\_ID" INT,

PRIMARY KEY("User\_ID"),

FOREIGN KEY("mgr\_ID") REFERENCES "User"("ID")

ON DELETE SET NULL

ON UPDATE CASCADE,

FOREIGN KEY("User\_ID") REFERENCES "User"("ID")

ON DELETE CASCADE

ON UPDATE CASCADE

);

CREATE TABLE "Manager"(

"User\_ID" INT NOT NULL,

"Area" VARCHAR(50) NOT NULL,

PRIMARY KEY("User\_ID"),

FOREIGN KEY("User\_ID") REFERENCES "User"("ID")

);

CREATE TABLE "Waiter"(

"User\_ID" INT NOT NULL,

"Hours" NUMERIC(7,2) NOT NULL,

"Type" VARCHAR(50) NOT NULL,

PRIMARY KEY("User\_ID"),

FOREIGN KEY("User\_ID") REFERENCES "User"("ID")

ON DELETE CASCADE

ON UPDATE CASCADE

);

CREATE TABLE "Cook"(

"User\_ID" INT NOT NULL,

"Specialty" VARCHAR(50),

"Type" VARCHAR(50) NOT NULL,

PRIMARY KEY("User\_ID"),

FOREIGN KEY("User\_ID") REFERENCES "User"("ID")

ON DELETE CASCADE

ON UPDATE CASCADE

);

CREATE TABLE "Menu"(

"Type" VARCHAR(50) NOT NULL,

"Available" BOOLEAN NOT NULL,

PRIMARY KEY("Type")

);

CREATE TABLE "Dish"(

"Dish\_ID" SERIAL NOT NULL,

"Available" BOOLEAN NOT NULL,

"Price" NUMERIC(5, 2) NOT NULL,

"Description" VARCHAR(255),

"Menu\_Type" VARCHAR(50),

PRIMARY KEY("Dish\_ID"),

FOREIGN KEY("Menu\_Type") REFERENCES "Menu"("Type")

ON DELETE CASCADE

ON UPDATE CASCADE

);

CREATE TABLE "Customer"(

"User\_ID" INT NOT NULL,

"TableNo" INT,

PRIMARY KEY("User\_ID"),

FOREIGN KEY("TableNo") REFERENCES "Table"("TableNo")

ON DELETE SET NULL

ON UPDATE CASCADE

);

CREATE TABLE "Order"(

"Order\_ID" SERIAL NOT NULL,

"User\_ID" INT NOT NULL,

"Date\_Time" TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY("Order\_ID"),

FOREIGN KEY("User\_ID") REFERENCES "User"("ID")

ON DELETE CASCADE

ON UPDATE CASCADE

);

CREATE TABLE "Transaction"(

"Transaction\_ID" SERIAL NOT NULL,

"Amount" NUMERIC(7,2) NOT NULL,

"Date\_Time" TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY("Transaction\_ID")

);

CREATE TABLE "Review"(

"User\_ID" INT NOT NULL,

"Review\_ID" SERIAL NOT NULL,

"Description" TEXT NOT NULL,

"Rating" INT NOT NULL,

"Dish\_ID" INT,

PRIMARY KEY("User\_ID", "Review\_ID"),

FOREIGN KEY("User\_ID") REFERENCES "User"("ID")

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY("Dish\_ID") REFERENCES "Dish"("Dish\_ID")

ON DELETE CASCADE

ON UPDATE CASCADE

);

CREATE TABLE "Ingredient"(

"Name" VARCHAR(50) NOT NULL,

"Price" NUMERIC(7,2) NOT NULL,

"Exp\_Date" DATE NOT NULL,

"Quantity" NUMERIC (5,2) NOT NULL,

PRIMARY KEY("Name")

);

CREATE TABLE "Customer\_Transaction"(

"User\_ID" INT NOT NULL,

"Transaction\_ID" INT NOT NULL,

PRIMARY KEY("User\_ID", "Transaction\_ID"),

FOREIGN KEY("User\_ID") REFERENCES "User"("ID")

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY("Transaction\_ID") REFERENCES "Transaction"("Transaction\_ID")

ON DELETE CASCADE

ON UPDATE CASCADE

);

CREATE TABLE "Order\_Table"(

"Order\_ID" INT NOT NULL,

"TableNo" INT NOT NULL,

PRIMARY KEY("Order\_ID", "TableNo"),

FOREIGN KEY("Order\_ID") REFERENCES "Order"("Order\_ID")

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY("TableNo") REFERENCES "Table"("TableNo")

ON DELETE CASCADE

ON UPDATE CASCADE

);

CREATE TABLE "Online\_Order"(

"Order\_ID" INT NOT NULL,

"Application" VARCHAR(50) NOT NULL,

PRIMARY KEY("Order\_ID"),

FOREIGN KEY("Order\_ID") REFERENCES "Order"("Order\_ID")

ON DELETE CASCADE

ON UPDATE CASCADE

);

CREATE TABLE "In\_Store\_Order"(

"Order\_ID" INT NOT NULL,

"waiter\_ID" INT NOT NULL,

PRIMARY KEY("Order\_ID"),

FOREIGN KEY("Order\_ID") REFERENCES "Order"("Order\_ID")

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY("waiter\_ID") REFERENCES "User"("ID")

ON DELETE CASCADE

ON UPDATE CASCADE

);

CREATE TABLE "Order\_Transaction"(

"Transaction\_ID" INT NOT NULL,

"Order\_ID" INT NOT NULL,

PRIMARY KEY("Transaction\_ID", "Order\_ID"),

FOREIGN KEY("Transaction\_ID") REFERENCES "Transaction"("Transaction\_ID")

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY("Order\_ID") REFERENCES "Order"("Order\_ID")

ON DELETE CASCADE

ON UPDATE CASCADE

);

CREATE TABLE "Order\_Dish"(

"Dish\_ID" INT NOT NULL,

"Order\_ID" INT NOT NULL,

PRIMARY KEY("Dish\_ID", "Order\_ID"),

FOREIGN KEY ("Dish\_ID") REFERENCES "Dish"("Dish\_ID")

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY ("Order\_ID") REFERENCES "Order"("Order\_ID")

ON DELETE CASCADE

ON UPDATE CASCADE

);

CREATE TABLE "Dish\_Ingredient"(

"Dish\_ID" INT NOT NULL,

"Ing\_Name" VARCHAR(50) NOT NULL,

PRIMARY KEY("Dish\_ID", "Ing\_Name"),

FOREIGN KEY("Dish\_ID") REFERENCES "Dish"("Dish\_ID")

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY("Ing\_Name") REFERENCES "Ingredient"("Name")

ON DELETE CASCADE

ON UPDATE CASCADE

);

CREATE TABLE "Ingredient\_Supplier"(

"Supplier" VARCHAR(50) NOT NULL,

"Ing\_Name" VARCHAR(50) NOT NULL,

PRIMARY KEY("Supplier", "Ing\_Name"),

FOREIGN KEY("Ing\_Name") REFERENCES "Ingredient"("Name")

ON DELETE CASCADE

ON UPDATE CASCADE

);

CREATE FUNCTION test()

RETURNS Table ("User\_ID" int, "TableNo" int, "ID" int, "Username" varchar, "Password" varchar, "FirstName" varchar, "MiddleName" varchar, "LastName" varchar, "GivenName" varchar, "Addr1" varchar, "Addr2" varchar, "Province" varchar, "PostalCode" varchar, "Sex" varchar, "Phone" varchar, "DOB" date, "Email" varchar)

AS $$

SELECT \*

FROM "Customer" AS c INNER JOIN "User" AS u ON c."User\_ID" = u."ID"

$$ LANGUAGE sql;

CREATE FUNCTION user\_number()

RETURNS BIGINT

AS $$

SELECT COUNT(\*) FROM "User"

$$ LANGUAGE sql;