CSD2206 Term Project F19

HINALI TEJANI

C0939251

Table of Contents

Company Overview

Company Name

GridleGrills

Company Description

GridleGrills sells variety of griddles & grill pans from various top-rated companies. They offer many sizes and shapes in pans, for all indoor grilling needs at everyday low prices. Along with best nonstick material, they also offer cast iron, square, round, double burner and reversible.

Type of Company

GridleGrills is a retailer, selling individual griddles & grill pans of a variety of styles.

Product

Griddles & Grill pans

Website Resources

- www.walmart.ca
- www.canadiantire.com
- www.homedepot.ca

Product Attributes

Attributes

- Product id (UID)
- Model
- Brand
- Type
- Serial Number
- Description
- Rating
- Color
- Diameter
- Handle Material
- Shape
- Price
- Material
- Dimensions
- weight

GridleGrills

Invoice

INVOICE # 28915
DATE May 29, 2019
CUSTOMER ID # 720
ASSOCIATE ID # 22 ASSOCIATE NAME J. Folger

1060 Yonge Street
Toronto, ON M4W 2L4
416-325-2739

Billing Details:

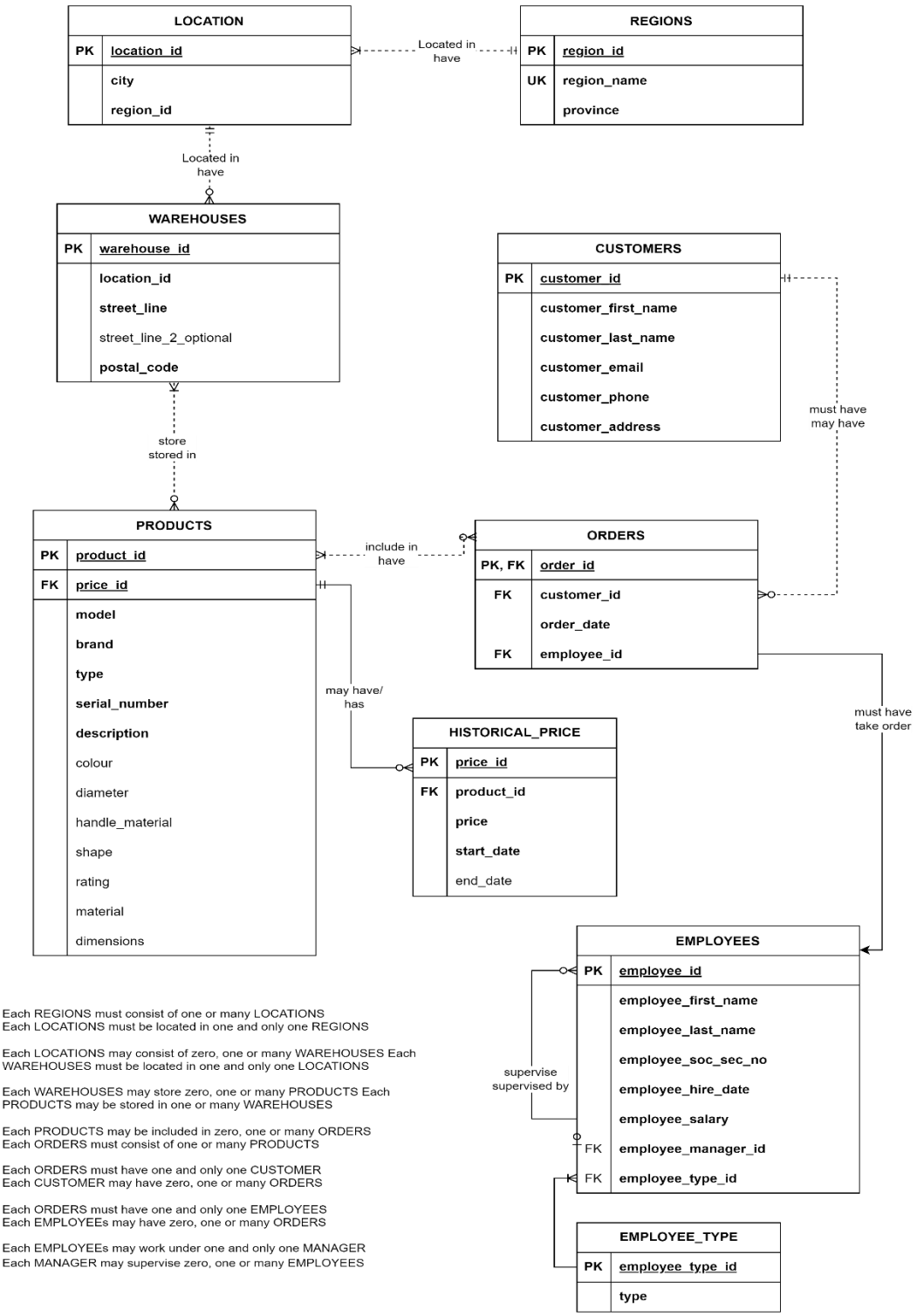
Dominick Mcdonnell
1080 Oxford Street E
London, ON N5Y 3L4
(519) 684-6631

PRODUCT ID	MODEL	BRAND	DESCRIPTION	QTY	UNIT PRICE	TOTAL
9001022	CE201	Ninja	BBQ Griddle	1	250.00	250.00
9001046	KF7150BK	Braun	3 Burner Gas Griddle	1	199.99	199.99
9001033	611247373590	Keurig	Cast Iron Griddle	1	68.00	68.00
9001014	KU5055746	Keurig	Electrical Grill	1	149.99	149.99
			SUBTOTAL			667.98
			SALES TAX			86.83
			SHIPPING & HANDLING			0.00
			TOTAL DUE			754.81

Thank you for shopping with us!

ER Diagrams

1. Many-to-many



Each REGIONS must consist of one or many LOCATIONS
Each LOCATIONS must be located in one and only one REGIONS

Each LOCATIONS may consist of zero, one or many WAREHOUSES Each WAREHOUSES must be located in one and only one LOCATIONS

Each WAREHOUSES may store zero, one or many PRODUCTS Each PRODUCTS may be stored in one or many WAREHOUSES

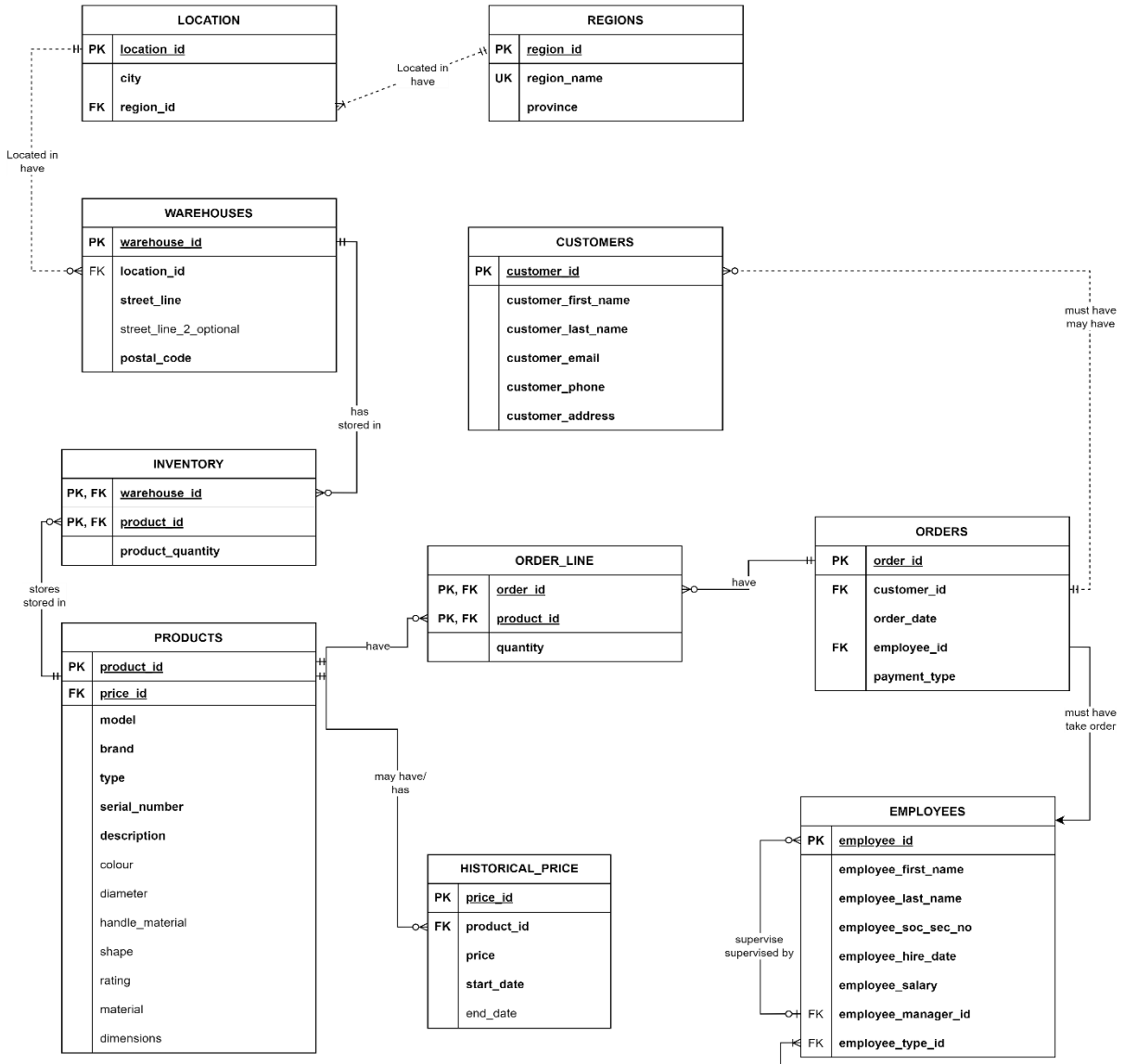
Each PRODUCTS may be included in zero, one or many ORDERS
Each ORDERS must consist of one or many PRODUCTS

Each ORDERS must have one and only one CUSTOMER
Each CUSTOMER may have zero, one or many ORDERS

Each ORDERS must have one and only one EMPLOYEES
Each EMPLOYEEs may have zero, one or many ORDERS

Each EMPLOYEEs may work under one and only one MANAGER
Each MANAGER may supervise zero, one or many EMPLOYEES

2. Many-to-many resolved



Each REGIONS must consist of one or many LOCATIONS
Each LOCATIONS must be located in one and only one REGIONS

Each LOCATIONS may consist of zero, one or many WAREHOUSES
Each WAREHOUSES must be located in one and only one LOCATIONS

Each WAREHOUSES may have zero, one or many INVENTORYs
Each INVENTORY must be stored in one and only one WAREHOUSES

Each INVENTORY must have one and only one PRODUCTS
Each PRODUCTS may be stored in one and only one INVENTORY

Each PRODUCTS may be included in zero, one or more ORDER_LINES
Each ORDER_LINES must include one and only one PRODUCTS

Each ORDER LINES must be included in one and only one ORDERs
Each ORDERS must include one or many ORDER LINES

Each ORDERS must have one and only one CUSTOMER
Each CUSTOMER may have zero, one or many ORDERS

Each ORDERS must have one and only one EMPLOYEES
Each EMPLOYEEs may have zero, one or many ORDERS

EACH EMPLOYEES may work under one and only one MANAGERS
Each MANAGER may supervise zero, one or many EMPLOYEES

Each PRODUCT may have one or many HISTORICAL_PRICES
Each HISTORICAL_PRICES must be has one and only one PRODUCT

Relational Model

PRODUCTS				
Key Type	**Optionality**	**Column Name**	**Data Type**	**Length**
PK	*	product_id	integer	
UK	*	model	varchar	20
	*	brand	varchar	20
	*	type	varchar	20
UK	*	serial_number	varchar	30
	*	description	varchar	60
	O	color	varchar	20
	O	diameter	decimal	3,2
	O	handle_material	varchar	30
	O	shape	varchar	20
	O	rating	decimal	3,2
	O	material	varchar	30
	O	dimensions	varchar	20

WAREHOUSES				
Key Type	**Optionality**	**Column Name**	**Data Type**	**Length**
PK	*	warehouse_id	integer	
FK	*	location_id	integer	
	*	street_address_line_1	varchar	30
	O	street_address_line_2	varchar	30
UK	*	Postal_code	varchar	7

INVENTORY				
Key Type	**Optionality**	**Column Name**	**Data Type**	**Length**
PK/FK	*	product_id	integer	
PK/FK	*	warehouse_id	integer	
	O	quantity	integer	

REGIONS				
Key Type	**Optionality**	**Column Name**	**Data Type**	**Length**
PK	*	region_id	integer	
UK	*	region_name	varchar	30
	*	province	varchar	30

LOCATIONS				
Key Type	**Optionality**	**Column Name**	**Data Type**	**Length**
PK	*	location_id	integer	
	*	city	varchar	30
FK	*	region_id	integer	

ORDER_LINES				
Key Type	**Optionality**	**Column Name**	**Data Type**	**Length**
PK/FK	*	order_id	integer	
PK/FK	*	product_id	integer	
	*	quantity	integer	
	*	price	decimal	6,2

ORDERS				
Key Type	**Optionality**	**Column Name**	**Data Type**	**Length**
PK	*	order_id	integer	
FK	*	customer_id	integer	
	*	order_date	date	
FK	*	employee_id	integer	

CUSTOMERS				
Key Type	**Optionality**	**Column Name**	**Data Type**	**Length**
PK	*	customer_id	integer	
	*	customer_first_name	varchar	20
	*	customer_last_name	varchar	20
UK	*	customer_email	varchar	30
	*	customer_phone	varchar	20
	*	customer_address	varchar	60

EMPLOYEES				
Key Type	**Optionality**	**Column Name**	**Data Type**	**Length**
PK	*	employee_id	integer	
	*	employee_first_name	varchar	20
	*	employee_last_name	varchar	20
	*	employee_soc_sec_no	integer	
	*	employee_hire_date	date	
	*	employee_salary	decimal	9,2
FK	*	employee_manager_id	integer	
	*	employee_type_id	varchar	20

HISTORICAL_PRICE				
Key Type	**Optionality**	**Column Name**	**Data Type**	**Length**
PK	*	price_id	integer	
FK	*	product_id	integer	

	*	price	decimal	6,2
	*	start_date	date	
	O	end_date	date	

PRODUCTS (**product_id**, model, brand, type, serial_number, description, color, diameter, handle_material, shape, rating, material, dimensions)

WAREHOUSES (**warehouse_id**, *location_id*, street_address_line_1, postal_code, street_address_line_2)

FK location_id → LOCATIONS

INVENTORY (***product_id***, ***warehouse_id***, quantity)

FK product_id → PRODUCTS

FK warehouse_id → WAREHOUSES

REGIONS (**region_id**, region_name, province)

LOCATIONS (**location_id**, city, *region_id*)

FK region_id → REGIONS

INVOICE(**invoice_id, *order_id*, *product_id***, quantity, price, payment_type, shipping_type)

FK order_id → ORDERS
FK product_id → PRODUCTS

ORDERS (**order_id**, *customer_id*, order_date, *employee_id*)

FK customer_id → CUSTOMERS
FK employee_id → EMPLOYEES

CUSTOMERS (**customer_id**, customer_first_name, customer_last_name, customer_email, customer_phone, customer_address)

EMPLOYEES (**employee_id**, employee_first_name, employee_last_name, employee_hire_date, employee_salary, employee_*manager_id*)

FK manager_id → EMPLOYEES

HISTORICAL_PRICE(**price_id, *product_id***, price, start_date, end_date)

FK product_id → PRODUCTS

Create Statements

```
CREATE TABLE products (
        product_id              INTEGER NOT NULL,
        model                   VARCHAR (20) NOT NULL,
        brand                   VARCHAR (20) NOT NULL,
        type                    VARCHAR (20) NOT NULL,
        serial_number           VARCHAR (30) NOT NULL,
        description             VARCHAR (60) NOT NULL,
        color                   VARCHAR (20) NOT NULL,
        diameter                DECIMAL (3,2),
        handle_material         VARCHAR (30) NOT NULL,
        shape                   VARCHAR (20) NOT NULL,
        rating                  DECIMAL (3,2),
        material                VARCHAR (30) NOT NULL,
        dimensions              VARCHAR (20)
);

CREATE TABLE warehouses (
        warehouse_id            INTEGER NOT NULL,
        location_id             INTEGER NOT NULL,
        street_address_line_1   VARCHAR(30) NOT NULL,
        street_address_line_2   VARCHAR(30) NOT NULL,
        postal_code             VARCHAR(7) NOT NULL
);

CREATE TABLE inventory (
        product_id              INTEGER NOT NULL,
        warehouse_id            INTEGER NOT NULL,
        quantity                INTEGER
);

CREATE TABLE regions (
        region_id               INTEGER NOT NULL,
        region_name             VARCHAR(30) NOT NULL,
        province                VARCHAR(30) NOT NULL
);

CREATE TABLE locations (
        location_id             INTEGER NOT NULL,
        city                    VARCHAR(30) NOT NULL,
        region_id               INTEGER NOT NULL,
```

```
);


CREATE  TABLE  employees (
      employee_id              INTEGER NOT NULL,
      employee_first_name      VARCHAR (20) NOT NULL,
      employee_last_name       VARCHAR (20) NOT NULL,
      employee_soc_sec_no      INTEGER NOT  NULL,
      employee_hire_date       DATE DEFAULT sysdate  NOT NULL,
      employee_salary          DECIMAL  (9,2) NOT  NULL,
      employee_manager_id      INTEGER
);

CREATE  TABLE  customers (
      customer_id              INTEGER NOT NULL,
      customer_first_name      VARCHAR (20) NOT NULL,
      customer_last_name       VARCHAR (20) NOT NULL,
      customer_email           VARCHAR (20),
      customer_phone           VARCHAR (20),
      customer_address         VARCHAR (60),
);

CREATE  TABLE  orders (
      order_id                 INTEGER NOT NULL,
      customer_id              INTEGER NOT NULL,
      employee_id              INTEGER NOT NULL,
      order_date               DATE DEFAULT sysdate  NOT NULL,
);

CREATE  TABLE  orders_lines (
      order_id                 INTEGER NOT NULL,
      product_id               INTEGER NOT NULL,
      quantity                 INTEGER NOT NULL,
      price                    DECIMAL  (6,2) NOT  NULL,
);

CREATE  TABLE  historical_price (
      price_id                 INTEGER NOT NULL,
      product_id               INTEGER NOT NULL,
      price                    DECIMAL  (6,2) NOT  NULL,
      quantity                 INTEGER NOT NULL,
      start_date               DATE NOT NULL,
      end_date                 DATE
);
```

Database Constraints

PRIMARY KEYS:

```
ALTER TABLE  warehouses
ADD CONSTRAINT warehouses_pk
PRIMARY KEY (warehouse_id);

ALTER TABLE  employees
ADD CONSTRAINT employees_pk
PRIMARY KEY (employee_id);

ALTER TABLE  PRODUCTS
ADD CONSTRAINT pk_products
PRIMARY KEY (product_id);

ALTER TABLE  INVENTORY
ADD CONSTRAINT pk_inventory
PRIMARY KEY (product_id, warehouse_id);

ALTER TABLE  REGIONS
ADD CONSTRAINT pk_regions
PRIMARY KEY (region_id);

ALTER TABLE  LOCATIONS
ADD CONSTRAINT pk_location
PRIMARY KEY (location_id);

ALTER TABLE  CUSTOMERS
ADD CONSTRAINT pk_customers
PRIMARY KEY (customer_id);

ALTER TABLE  ORDERS
ADD CONSTRAINT pk_orders
PRIMARY KEY (order_id);

ALTER TABLE  ORDER_LINES
ADD CONSTRAINT pk_order_line
PRIMARY KEY (order_id, product_id);

ALTER TABLE  HISTORICAL_PRICE
ADD CONSTRAINT pk_historical_price
PRIMARY KEY (price_id);
```

```
ALTER TABLE  EMPLOYEE_TYPE
ADD CONSTRAINT pk_employee_type
PRIMARY KEY (employee_type_id);
```

FORIEGN KEYS:

```
ALTER TABLE  EMPLOYEES
ADD CONSTRAINT fk_employees_employee_type
FOREIGN KEY (employee_type_id) REFERENCES EMPLOYEE_TYPE(employee_type_id);

ALTER TABLE  EMPLOYEES
ADD CONSTRAINT fk_employees_manager
FOREIGN KEY (employee_manager_id) REFERENCES EMPLOYEES(employee_id);

ALTER TABLE  WAREHOUSES
ADD CONSTRAINT fk_warehouses_location
FOREIGN KEY (location_id) REFERENCES LOCATIONS(location_id);

ALTER TABLE  LOCATIONS
ADD CONSTRAINT fk_location_regions
FOREIGN KEY (region_id) REFERENCES REGIONS(region_id);

ALTER TABLE  INVENTORY
ADD CONSTRAINT fk_inventory_warehouse
FOREIGN KEY (warehouse_id) REFERENCES WAREHOUSES(warehouse_id);

ALTER TABLE  INVENTORY
ADD CONSTRAINT fk_inventory_product
FOREIGN KEY (product_id) REFERENCES PRODUCTS(product_id);

ALTER TABLE  ORDERS
ADD CONSTRAINT fk_orders_customer
FOREIGN KEY (customer_id) REFERENCES CUSTOMERS(customer_id);

ALTER TABLE  ORDERS
ADD CONSTRAINT fk_orders_employee
FOREIGN KEY (employee_id) REFERENCES EMPLOYEES(employee_id);

ALTER TABLE  ORDER_LINES
ADD CONSTRAINT fk_order_line_product
FOREIGN KEY (product_id) REFERENCES PRODUCTS(product_id);
```

```
ALTER TABLE  ORDER_LINES
ADD CONSTRAINT fk_order_line_order
FOREIGN KEY (order_id) REFERENCES ORDERS(order_id);

ALTER TABLE  PRODUCTS
ADD CONSTRAINT fk_order_line_order
FOREIGN KEY (price_id) REFERENCES HISTORICAL_PRICE(price_id);
```

UNIQUE KEYS:

```
ALTER TABLE  CUSTOMERS
ADD CONSTRAINT uk_customers_email UNIQUE (customer_email);

ALTER TABLE  PRODUCTS
ADD CONSTRAINT uk_MODEL UNIQUE (MODEL);

ALTER TABLE  REGIONS
ADD CONSTRAINT uk_region_name UNIQUE (region_name);

ALTER TABLE  REGIONS
ADD CONSTRAINT uk_province UNIQUE (province);

ALTER TABLE  employees
ADD CONSTRAINT uk_soc_sec_no UNIQUE (employee_soc_sec_no);
```

BUSINESS CONSTRAINS:

```
ALTER TABLE  employees
ADD CONSTRAINT employees_soc_sec_no CHECK (employee_soc_sec_no BETWEEN 111111111
AND 999999999);

ALTER TABLE  order_lines
ADD CONSTRAINT order_items_quantity CHECK (quantity > 0);
```

Insert Statements

PRODUCT TABLE:

```
INSERT ALL
  INTO PRODUCTS (product_id, model, brand, type, serial_number, description, color, diameter,
handle_material, shape, rating, material, dimensions)

      VALUES (1, 'L9OG3', 'Lodge', 'Griddle', 'L9OG3', 'Cast iron griddle, excellent heat retention',
'Black', 10.5, 'Cast iron', 'Round', 4.8, 'Cast iron', '10.5 x 10.5 x 0.5 in')

  INTO PRODUCTS (product_id, model, brand, type, serial_number, description, color, diameter,
handle_material, shape, rating, material, dimensions)

      VALUES (2, 'B036S3', 'T-fal', 'Griddle', 'B036S3', 'Hard anodized nonstick griddle, durable',
'Black', 11, 'Plastic', 'Rectangular', 4.6, 'Hard anodized aluminum', '11 x 11 x 0.5 in')

  INTO PRODUCTS (product_id, model, brand, type, serial_number, description, color, diameter,
handle_material, shape, rating, material, dimensions)

      VALUES (3, '622-24', 'Cuisinart', 'Griddle', '622-24', 'Nonstick griddle with cool-touch handle',
'Black', 24, 'Stainless steel', 'Rectangular', 4.7, 'Hard-anodized aluminum', '24 x 12 x 0.5 in')

  INTO PRODUCTS (product_id, model, brand, type, serial_number, description, color, diameter,
handle_material, shape, rating, material, dimensions)

      VALUES (4, '176-52', 'Calphalon', 'Griddle', '176-52', 'Nonstick griddle with even heat
distribution', 'Black', 11, 'Stainless steel', 'Rectangular', 4.5, 'Hard-anodized aluminum', '11 x 11 x
0.5 in')

  INTO PRODUCTS (product_id, model, brand, type, serial_number, description, color, diameter,
handle_material, shape, rating, material, dimensions)

      VALUES (5, '601007', 'All-Clad', 'Griddle', '601007', 'Stainless steel griddle, even heat
distribution', 'Silver', 11, 'Stainless steel', 'Rectangular', 4.6, 'Stainless steel', '11 x 11 x 0.5 in')

  INTO PRODUCTS (product_id, model, brand, type, serial_number, description, color, diameter,
handle_material, shape, rating, material, dimensions)

      VALUES (6, 'LDP3', 'Lodge', 'Grill/Griddle', 'LDP3', 'Reversible grill and griddle, cast iron', 'Black',
20, 'Cast iron', 'Rectangular', 4.7, 'Cast iron', '20 x 10.5 x 0.75 in')

  INTO PRODUCTS (product_id, model, brand, type, serial_number, description, color, diameter,
handle_material, shape, rating, material, dimensions)

      VALUES (7, '17602', 'Rachael Ray', 'Griddle', '17602', 'Durable nonstick griddle with
comfortable handle', 'Orange', 11, 'Plastic', 'Rectangular', 4.5, 'Hard-anodized aluminum', '11 x 11
x 0.5 in')

  INTO PRODUCTS (product_id, model, brand, type, serial_number, description, color, diameter,
```

```
handle_material, shape, rating, material, dimensions)
      VALUES (8, 'E93808', 'T-fal', 'Griddle', 'E93808', 'Professional nonstick griddle with heat
indicator', 'Black', 12, 'Plastic', 'Rectangular', 4.6, 'Hard anodized aluminum', '12 x 12 x 0.5 in')

  INTO PRODUCTS (product_id, model, brand, type, serial_number, description, color, diameter,
handle_material, shape, rating, material, dimensions)

      VALUES (9, 'VGRB', 'Victoria', 'Grill/Griddle', 'VGRB', 'Reversible grill and griddle, cast iron',
'Black', 17, 'Cast iron', 'Rectangular', 4.8, 'Cast iron', '17 x 9 x 0.75 in')

  INTO PRODUCTS (product_id, model, brand, type, serial_number, description, color, diameter,
handle_material, shape, rating, material, dimensions)

      VALUES (10, 'CC002835-001', 'GreenPan', 'Griddle', 'CC002835-001', 'Ceramic nonstick griddle,
PFAS-free', 'Gray', 11, 'Bakelite', 'Rectangular', 4.6, 'Ceramic coating', '11 x 11 x 0.5 in')
SELECT 1 FROM DUAL;
```

WAREHOUSE TABLE:

```
INSERT ALL
  INTO warehouses (warehouse_id, location_id, street_address_line_1, street_address_line_2,
postal_code)
      VALUES (1, 1, '123 King St W', 'Suite 200', 'M5H 3T9')
  INTO warehouses (warehouse_id, location_id, street_address_line_1, street_address_line_2,
postal_code)
      VALUES (2, 2, '456 Queen St E', 'Suite 300', 'M5A 1T7')
  INTO warehouses (warehouse_id, location_id, street_address_line_1, street_address_line_2,
postal_code)
      VALUES (3, 3, '789 Dundas St W', 'Suite 400', 'M6J 1X5')
  INTO warehouses (warehouse_id, location_id, street_address_line_1, street_address_line_2,
postal_code)
      VALUES (4, 4, '101 Bathurst St', 'Suite 500', 'M5V 2P3')
  INTO warehouses (warehouse_id, location_id, street_address_line_1, street_address_line_2,
postal_code)
      VALUES (5, 5, '202 Bloor St E', 'Suite 600', 'M4W 1A8')
  INTO warehouses (warehouse_id, location_id, street_address_line_1, street_address_line_2,
postal_code)
      VALUES (6, 6, '303 Richmond St W', 'Suite 700', 'M5V 1X3')
  INTO warehouses (warehouse_id, location_id, street_address_line_1, street_address_line_2,
postal_code)
      VALUES (7, 7, '404 College St', 'Suite 800', 'M5T 1S6')
  INTO warehouses (warehouse_id, location_id, street_address_line_1, street_address_line_2,
postal_code)
      VALUES (8, 8, '505 Yonge St', 'Suite 900', 'M4Y 1X6')
  INTO warehouses (warehouse_id, location_id, street_address_line_1, street_address_line_2,
postal_code)
      VALUES (9, 9, '606 Front St W', 'Suite 1000', 'M5V 1E3')
  INTO warehouses (warehouse_id, location_id, street_address_line_1, street_address_line_2,
postal_code)
      VALUES (10, 10, '707 Gerrard St E', 'Suite 1100', 'M4M 1Y1')
SELECT 1 FROM dual;
```

REGION TABLE:

```
INSERT ALL
  INTO regions (region_id, region_name, province)
      VALUES (101, 'Greater Toronto Area', 'Ontario')
  INTO regions (region_id, region_name, province)
      VALUES (102, 'Montreal Area', 'Quebec')
  INTO regions (region_id, region_name, province)
      VALUES (103, 'Vancouver Area', 'British Columbia')
  INTO regions (region_id, region_name, province)
      VALUES (104, 'Calgary Area', 'Alberta')
  INTO regions (region_id, region_name, province)
      VALUES (105, 'Winnipeg Area', 'Manitoba')
  INTO regions (region_id, region_name, province)
      VALUES (106, 'Regina Area', 'Saskatchewan')
  INTO regions (region_id, region_name, province)
      VALUES (107, 'Halifax Area', 'Nova Scotia')
  INTO regions (region_id, region_name, province)
      VALUES (108, 'St. John Area', 'Newfoundland and Labrador')
  INTO regions (region_id, region_name, province)
      VALUES (109, 'Charlottetown Area', 'Prince Edward Island')
  INTO regions (region_id, region_name, province)
      VALUES (110, 'Fredericton Area', 'New Brunswick')
SELECT 1 FROM dual;
```

LOCATION TABLE:

```
INSERT ALL
  INTO locations (location_id, city, region_id)
      VALUES (1, 'Toronto', 101)
  INTO locations (location_id, city, region_id)
      VALUES (2, 'Ottawa', 102)
  INTO locations (location_id, city, region_id)
      VALUES (3, 'Mississauga', 103)
  INTO locations (location_id, city, region_id)
      VALUES (4, 'Brampton', 104)
  INTO locations (location_id, city, region_id)
      VALUES (5, 'Hamilton', 105)
  INTO locations (location_id, city, region_id)
      VALUES (6, 'London', 106)
  INTO locations (location_id, city, region_id)
      VALUES (7, 'Windsor', 107)
  INTO locations (location_id, city, region_id)
      VALUES (8, 'Kitchener', 108)
  INTO locations (location_id, city, region_id)
      VALUES (9, 'Guelph', 109)
  INTO locations (location_id, city, region_id)
      VALUES (10, 'Oshawa', 110)
SELECT 1 FROM dual;
```

INVENTORY TABLE:

```
INSERT ALL
  INTO inventory (product_id, warehouse_id, quantity)
      VALUES (1, 1, 100)
  INTO inventory (product_id, warehouse_id, quantity)
      VALUES (2, 2, 150)
  INTO inventory (product_id, warehouse_id, quantity)
      VALUES (3, 2, 200)
  INTO inventory (product_id, warehouse_id, quantity)
      VALUES (4, 2, 250)
  INTO inventory (product_id, warehouse_id, quantity)
      VALUES (5, 3, 300)
  INTO inventory (product_id, warehouse_id, quantity)
      VALUES (6, 3, 350)
  INTO inventory (product_id, warehouse_id, quantity)
      VALUES (7, 4, 400)
  INTO inventory (product_id, warehouse_id, quantity)
      VALUES (8, 4, 450)
  INTO inventory (product_id, warehouse_id, quantity)
      VALUES (9, 5, 500)
  INTO inventory (product_id, warehouse_id, quantity)
      VALUES (10, 5, 550)
SELECT 1 FROM dual;
```

EMPLOYEES TABLE:

```
INSERT ALL
  INTO employees (employee_id, employee_first_name, employee_last_name,
employee_soc_sec_no, employee_hire_date, employee_salary, employee_manager_id)
      VALUES (1, 'John', 'Doe', 123456789, TO_DATE('2022-01-15', 'YYYY-MM-DD'), 75000.00, NULL)

  INTO employees (employee_id, employee_first_name, employee_last_name,
employee_soc_sec_no, employee_hire_date, employee_salary, employee_manager_id)
      VALUES (2, 'Jane', 'Smith', 234567890, TO_DATE('2021-11-23', 'YYYY-MM-DD'), 80000.00, 1)

  INTO employees (employee_id, employee_first_name, employee_last_name,
employee_soc_sec_no, employee_hire_date, employee_salary, employee_manager_id)
      VALUES (3, 'Emily', 'Johnson', 345678901, TO_DATE('2020-05-30', 'YYYY-MM-DD'), 70000.00,
1)

  INTO employees (employee_id, employee_first_name, employee_last_name,
employee_soc_sec_no, employee_hire_date, employee_salary, employee_manager_id)
      VALUES (4, 'Michael', 'Williams', 456789012, TO_DATE('2019-08-15', 'YYYY-MM-DD'),
72000.00, 2)

  INTO employees (employee_id, employee_first_name, employee_last_name,
employee_soc_sec_no, employee_hire_date, employee_salary, employee_manager_id)
      VALUES (5, 'Linda', 'Brown', 567890123, TO_DATE('2023-03-01', 'YYYY-MM-DD'), 68000.00, 2)

  INTO employees (employee_id, employee_first_name, employee_last_name,
employee_soc_sec_no, employee_hire_date, employee_salary, employee_manager_id)
```

```
    VALUES (6, 'James', 'Davis', 678901234, TO_DATE('2021-06-10', 'YYYY-MM-DD'), 71000.00, 3)

  INTO employees (employee_id, employee_first_name, employee_last_name,
employee_soc_sec_no, employee_hire_date, employee_salary, employee_manager_id)
    VALUES (7, 'Patricia', 'Miller', 789012345, TO_DATE('2022-07-22', 'YYYY-MM-DD'), 69000.00, 3)

  INTO employees (employee_id, employee_first_name, employee_last_name,
employee_soc_sec_no, employee_hire_date, employee_salary, employee_manager_id)
    VALUES (8, 'Robert', 'Wilson', 890123456, TO_DATE('2020-12-05', 'YYYY-MM-DD'), 73000.00,
4)

  INTO employees (employee_id, employee_first_name, employee_last_name,
employee_soc_sec_no, employee_hire_date, employee_salary, employee_manager_id)
    VALUES (9, 'Susan', 'Moore', 901234567, TO_DATE('2018-09-18', 'YYYY-MM-DD'), 67000.00, 4)

  INTO employees (employee_id, employee_first_name, employee_last_name,
employee_soc_sec_no, employee_hire_date, employee_salary, employee_manager_id)
    VALUES (10, 'David', 'Taylor', 123098456, TO_DATE('2019-11-30', 'YYYY-MM-DD'), 75000.00,
NULL)

SELECT 1 FROM dual;
```

CUSTOMER TABLE:

```
INSERT ALL
  INTO customers (customer_id, customer_first_name, customer_last_name, customer_email,
customer_phone, customer_address)
    VALUES (1, 'Alice', 'Johnson', 'alice.johnson@example.com', '123-456-7890', '123 Maple St,
Toronto, ON')
  INTO customers (customer_id, customer_first_name, customer_last_name, customer_email,
customer_phone, customer_address)
    VALUES (2, 'Bob', 'Smith', 'bob.smith@example.com', '234-567-8901', '456 Oak St, Montreal,
QC')
  INTO customers (customer_id, customer_first_name, customer_last_name, customer_email,
customer_phone, customer_address)
    VALUES (3, 'Carol', 'Williams', 'carol.williams@example.com', '345-678-9012', '789 Pine St,
Vancouver, BC')
  INTO customers (customer_id, customer_first_name, customer_last_name, customer_email,
customer_phone, customer_address)
    VALUES (4, 'David', 'Brown', 'david.brown@example.com', '456-789-0123', '101 Birch St,
Calgary, AB')
  INTO customers (customer_id, customer_first_name, customer_last_name, customer_email,
customer_phone, customer_address)
    VALUES (5, 'Eve', 'Davis', 'eve.davis@example.com', '567-890-1234', '202 Cedar St, Winnipeg,
MB')
  INTO customers (customer_id, customer_first_name, customer_last_name, customer_email,
customer_phone, customer_address)
    VALUES (6, 'Frank', 'Miller', 'frank.miller@example.com', '678-901-2345', '303 Elm St, Halifax,
NS')
  INTO customers (customer_id, customer_first_name, customer_last_name, customer_email,
customer_phone, customer_address)
```

```
    VALUES (7, 'Grace', 'Wilson', 'grace.wilson@example.com', '789-012-3456', '404 Spruce St, Regina, SK')
  INTO customers (customer_id, customer_first_name, customer_last_name, customer_email, customer_phone, customer_address)
    VALUES (8, 'Hank', 'Moore', 'hank.moore@example.com', '890-123-4567', '505 Fir St, St. John\'s, NL')
  INTO customers (customer_id, customer_first_name, customer_last_name, customer_email, customer_phone, customer_address)
    VALUES (9, 'Ivy', 'Taylor', 'ivy.taylor@example.com', '901-234-5678', '606 Poplar St, Charlottetown, PE')
  INTO customers (customer_id, customer_first_name, customer_last_name, customer_email, customer_phone, customer_address)
    VALUES (10, 'Jack', 'Anderson', 'jack.anderson@example.com', '012-345-6789', '707 Willow St, Fredericton, NB')
SELECT 1 FROM dual;
```

ORDERS TABLE:

```
INSERT ALL
  INTO orders (order_id, customer_id, employee_id, order_date)
    VALUES (1, 1, 1, TO_DATE('2023-07-01', 'YYYY-MM-DD'))
  INTO orders (order_id, customer_id, employee_id, order_date)
    VALUES (2, 2, 2, TO_DATE('2023-07-05', 'YYYY-MM-DD'))
  INTO orders (order_id, customer_id, employee_id, order_date)
    VALUES (3, 3, 3, TO_DATE('2023-07-10', 'YYYY-MM-DD'))
  INTO orders (order_id, customer_id, employee_id, order_date)
    VALUES (4, 4, 4, TO_DATE('2023-07-15', 'YYYY-MM-DD'))
  INTO orders (order_id, customer_id, employee_id, order_date)
    VALUES (5, 5, 5, TO_DATE('2023-07-20', 'YYYY-MM-DD'))
  INTO orders (order_id, customer_id, employee_id, order_date)
    VALUES (6, 6, 6, TO_DATE('2023-07-25', 'YYYY-MM-DD'))
  INTO orders (order_id, customer_id, employee_id, order_date)
    VALUES (7, 7, 7, TO_DATE('2023-07-30', 'YYYY-MM-DD'))
  INTO orders (order_id, customer_id, employee_id, order_date)
    VALUES (8, 8, 8, TO_DATE('2023-08-01', 'YYYY-MM-DD'))
  INTO orders (order_id, customer_id, employee_id, order_date)
    VALUES (9, 9, 9, TO_DATE('2023-08-05', 'YYYY-MM-DD'))
  INTO orders (order_id, customer_id, employee_id, order_date)
    VALUES (10, 10, 10, TO_DATE('2023-08-10', 'YYYY-MM-DD'))
SELECT 1 FROM dual;
```

ORDER_LINES TABLE:

```
INSERT ALL
  INTO order_lines (order_id, product_id, quantity, price)
    VALUES (1, 1, 2, 29.99)
  INTO order_lines (order_id, product_id, quantity, price)
    VALUES (1, 2, 1, 45.50)
  INTO order_lines (order_id, product_id, quantity, price)
    VALUES (2, 3, 3, 15.99)
```

```
  INTO order_lines (order_id, product_id, quantity, price)
     VALUES (3, 4, 5, 25.00)
  INTO order_lines (order_id, product_id, quantity, price)
     VALUES (4, 5, 2, 49.99)
  INTO order_lines (order_id, product_id, quantity, price)
     VALUES (5, 6, 1, 32.75)
  INTO order_lines (order_id, product_id, quantity, price)
     VALUES (6, 7, 4, 10.50)
  INTO order_lines (order_id, product_id, quantity, price)
     VALUES (7, 8, 3, 20.00)
  INTO order_lines (order_id, product_id, quantity, price)
     VALUES (8, 9, 2, 18.00)
  INTO order_lines (order_id, product_id, quantity, price)
     VALUES (9, 10, 1, 55.99)
SELECT 1 FROM dual;
```

Constraint Testing

```
-- CONSTRAINT TESTING

-- Constraint Test 1
-- Description: Confirm primary key constraint on product_id column in products table
-- Expected result: Insert fails with duplicate key error
-- Action
    INSERT INTO products VALUES (1, 'L9OG3', 'Lodge', 'Griddle', 'L9OG3', 'Cast iron griddle,
excellent heat retention', 'Black', 10.5, 'Cast iron', 'Round', 4.8, 'Cast iron', '10.5 x 10.5 x 0.5 in');
-- Result:
-- unique constraint (WKSP_HINALILAMBTON.PK_PRODUCTS) violated.

-- Constraint Test 2
-- Description: Confirm primary key constraint on ORDER_LINE column in order_line table
-- Expected result: Insert fails with duplicate key error
-- Action
    INSERT INTO order_lines VALUES (1, 1, 2, 29.99);
-- Result:
-- ORA-00001: unique constraint (WKSP_HINALILAMBTON.PK_ORDER_LINE) violated

-- Constraint Test 3
-- Description: Confirm primary key constraint on inventory column in inventorytable
-- Expected result: Insert fails with duplicate key error
-- Action
    INSERT INTO inventory VALUES (1, 1, 100);
-- Result:
-- ORA-00001: unique constraint (WKSP_HINALILAMBTON.PK_INVENTORY) violated

-- Constraint Test 4
-- Description: Can not insert NULL value
-- Expected result: Insert fails, operation not allowed
-- Action
    INSERT INTO employees VALUES (1, 'John', 'Doe', 123456789, TO_DATE('2022-01-15',
'YYYY-MM-DD'), 75000.00, NULL);
-- Result:
-- ORA-01400: cannot insert NULL into
("WKSP_HINALILAMBTON"."EMPLOYEES"."EMPLOYEE_TYPE_ID")
```

```
-- Constraint Test 5
-- Description: Insert value larger than defined in data type for POSTAL_CODE
-- Expected result: Operation not allowed for large value constraint
-- Action
    INSERT INTO warehouses VALUES (1, 1, '123 King St W', 'Suite 200', 'M5H 3T39');
-- Result:
-- ORA-12899: value too large for column
"WKSP_HINALILAMBTON"."WAREHOUSES"."POSTAL_CODE" (actual: 8, maximum: 7)
```

Views

PRODV1

```
CREATE OR REPLACE VIEW prodV1 AS
SELECT
    p.product_id, p.model, p.brand, p.type, p.description, p.color, p.diameter,
    p.handle_material, p.shape, p.rating, p.material, p.dimensions, i.warehouse_id, i.quantity
FROM
    products p
LEFT JOIN
    inventory i ON p.PRODUCT_ID = i.PRODUCT_ID;
```

INVENTORY_SUMMARY

```
CREATE OR REPLACE VIEW inventory_summary AS
SELECT
    p.product_id, p.model, p.brand,
    SUM(i.quantity) AS total_quantity
FROM
    products p
INNER JOIN
    inventory i ON p.product_id = i.product_id
GROUP BY
      p.product_id, p.model, p.brand;
```