

CSD2206 Term Project F19

MITCH CARPENTER
C0750361

Table of Contents

Company Overview.....	2
Company Name	2
Company Description	2
Type of Company	2
Product	2
Website Resources.....	2
Product Attributes.....	3
Attributes.....	3
Invoice.....	4
ER Diagrams.....	5
Relational Model.....	7
Create Statements.....	10
Database Constraints.....	12
Insert Statements.....	15
Constraint Testing.....	27
Views	31
PRODV1.....	31
PRODV2.....	31

Company Overview

Company Name

JustBrewIt

Company Description

JustBrewIt sells coffee makers from various top-rated companies. They offer both single serve and traditional drip style coffee makers. Along with regular coffee makers, they also offer espresso, latte, and cappuccino machines.

Type of Company

JustBrewIt is a retailer, selling individual coffee makers of a variety of styles.

Product

Coffee makers (coffee, espresso, latte, cappuccino)

Website Resources

- www.walmart.ca
- www.bedbathandbeyond.ca
- www.hamiltonbeach.com

Product Attributes

Attributes

- Product id (UID)
- Model
- Brand
- Type
- Serial Number
- Description
- Rating
- Colour
- Cup Capacity
- Programmable
- Quick Prep
- Small Space
- Price

JustBrewIt

Invoice

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

INVOICE # 28915
DATE September 26, 2019
CUSTOMER ID # 720
ASSOCIATE ID # 22
ASSOCIATE NAME J. Folger

BILL TO:

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

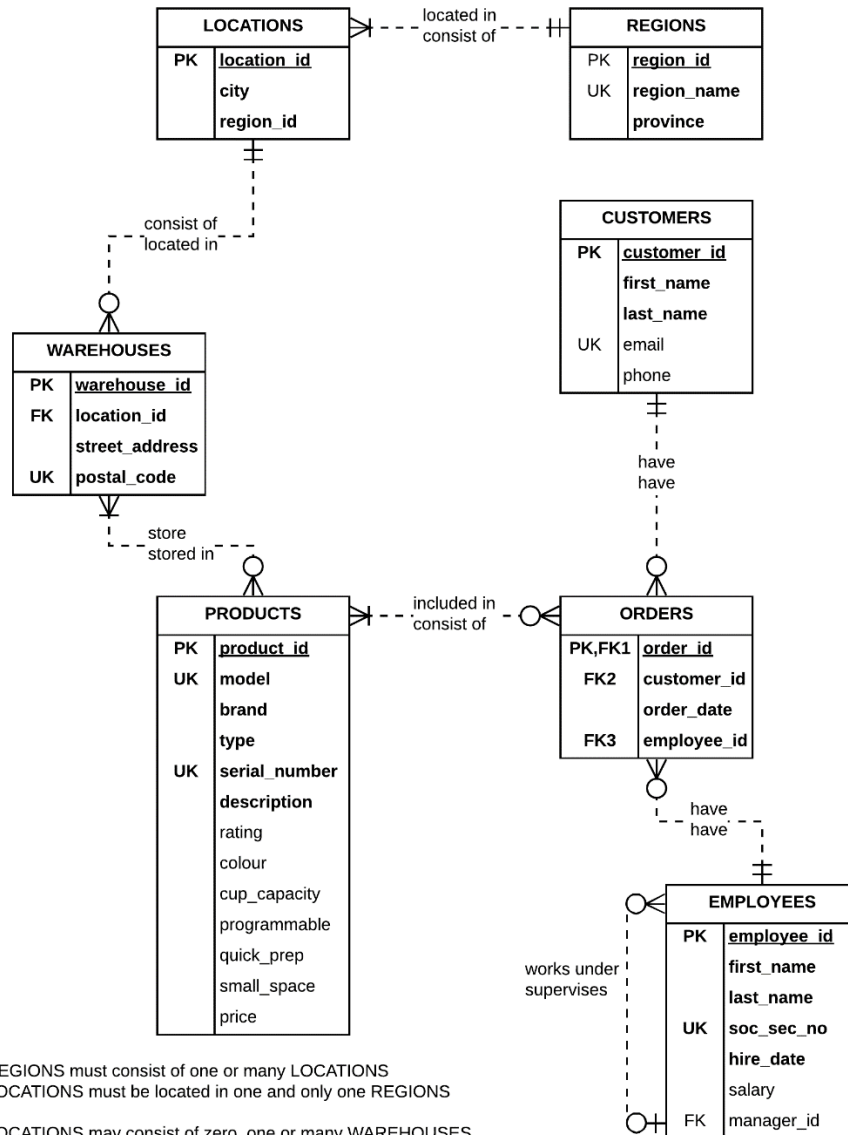
PRODUCT ID	MODEL	BRAND	DESCRIPTION	QTY	UNIT PRICE	TOTAL
9001022	CE201	Ninja	12-Cup Coffee Maker	2	129.19	258.38
9001046	KF7150BK	Braun	Brew Sense Drip Maker	1	99.99	99.99
9001033	611247373590	Keurig	K-Mini Single Serv	1	68.00	68.00
9001014	KU5055746	Keurig	K200 Brewing System	1	149.99	149.99
9001003	BVSBTF6	Sunbeam	5- Cup Coffee Maker - White	3	18.98	56.94
9001019	DCC3400	Cuisinart	12-Cup Programmable	2	145.27	290.54
SUBTOTAL						923.84
SALES TAX						120.10
SHIPPING & HANDLING						0.00
TOTAL DUE						1043.94

Thank you for your business!

Mitch Carpenter
C0750361

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

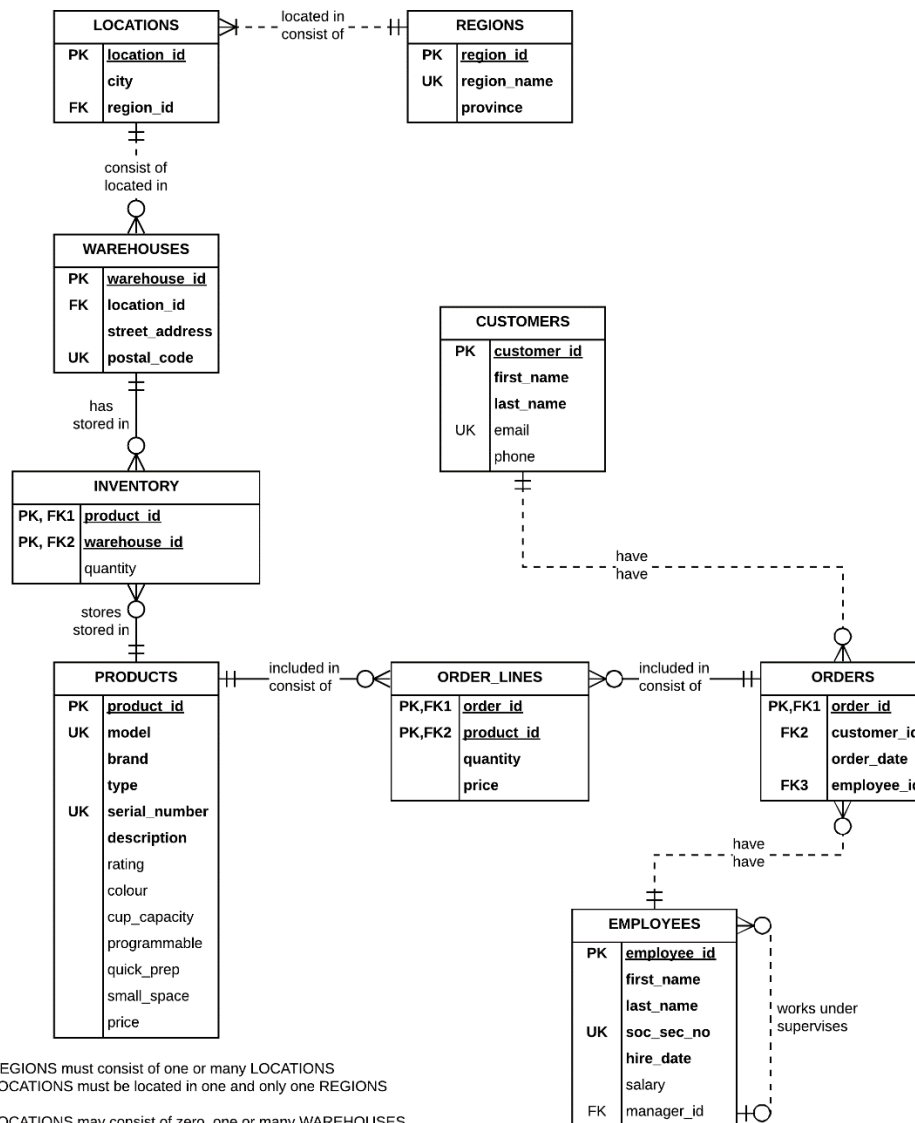
Each PRODUCTS may be included in zero, one or many ORDERS
 Each ORDERS must consist of one or many PRODUCTS

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 MANAGER
 Each MANAGER may supervise zero, one or many EMPLOYEES

Relational Model

PRODUCTS				
Key Type	Optionality	Column Name	Data Type	Length
pk	*	product_id	integer	
uk	*	model	varchar	20
	*	brand	varchar	20
	*	brew_type	varchar	20
uk	*	serial_number	varchar	30
	*	item_description	varchar	60
	0	rating	decimal	3,2
	0	colour	varchar	20
	0	cup_capacity	integer	
	0	programmable	bit	
	0	quick_prep	bit	
	0	small_space	bit	
	0	price	decimal	6,2

WAREHOUSES				
Key Type	Optionality	Column Name	Data Type	Length
pk	*	warehouse_id	integer	
fk	*	location_id	integer	
	*	street_address	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	
	0	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	
	*	first_name	varchar	20
	*	last_name	varchar	20
uk	0	email	varchar	30
	0	phone	varchar	20

EMPLOYEES				
Key Type	Optionality	Column Name	Data Type	Length
pk	*	employee_id	integer	
	*	first_name	varchar	20
	*	last_name	varchar	20
uk	*	soc_sec_no	integer	
	*	hire_date	date	
	0	salary	decimal	9,2
fk	0	manager_id	integer	

PRODUCTS (product_id, model, brand, type, serial_number, description, rating, colour, cup_capacity, programmable, quick_prep, small_space, price)

WAREHOUSES (warehouse_id, location_id, street_address, postal_code)

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

ORDER_LINES (order_id, product_id, quantity, price)

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, first_name, last_name, email, phone)

EMPLOYEES (employee_id, first_name, last_name, soc_sec_no, hire_date, salary,
manager_id)

FK manager_id → EMPLOYEES

Create Statements

```
CREATE TABLE jbi_products (  
    product_id      INTEGER NOT NULL,  
    model           VARCHAR (30) NOT NULL,  
    brand            VARCHAR (20) NOT NULL,  
    brew_type        VARCHAR (20) NOT NULL,  
    serial_number    VARCHAR (30) NOT NULL,  
    item_description VARCHAR (60) NOT NULL,  
    rating           DECIMAL (3,2),  
    colour           VARCHAR (20) NOT NULL,  
    cup_capacity      INTEGER,  
    programmable     NUMERIC,  
    quick_prep       NUMERIC,  
    small_space      NUMERIC,  
    price            DECIMAL (5,2)  
);  
  
CREATE TABLE jbi_warehouses (  
    warehouse_id     INTEGER NOT NULL,  
    location_id       INTEGER NOT NULL,  
    street_address    VARCHAR(30) NOT NULL,  
    postal_code       VARCHAR(7) NOT NULL  
);  
  
CREATE TABLE jbi_inventory (  
    product_id       INTEGER NOT NULL,  
    warehouse_id     INTEGER NOT NULL,  
    quantity         INTEGER  
);  
  
CREATE TABLE jbi_employees (  
    employee_id       INTEGER NOT NULL,  
    first_name        VARCHAR (20) NOT NULL,  
    last_name         VARCHAR (20) NOT NULL,  
    soc_sec_no        INTEGER NOT NULL,  
    hire_date         DATE NOT NULL DEFAULT CURRENT_DATE,  
    salary            DECIMAL (9,2) NOT NULL,  
    manager_id        INTEGER  
);
```

```

CREATE TABLE jbi_customers (
  customer_id    INTEGER NOT NULL,
  first_name     VARCHAR (20) NOT NULL,
  last_name      VARCHAR (20) NOT NULL,
  email          VARCHAR (30),
  phone          VARCHAR (20)
);

CREATE TABLE jbi_orders (
  order_id       INTEGER      NOT NULL,
  customer_id    INTEGER      NOT NULL,
  order_date     DATE         NOT NULL DEFAULT CURRENT_DATE,
  employee_id    INTEGER
);

CREATE TABLE jbi_locations (
  location_id    INTEGER      NOT NULL,
  city           VARCHAR(30) NOT NULL,
  region_id     INTEGER      NOT NULL
);

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

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

```

Database Constraints

```
ALTER TABLE jbi_warehouses
ADD CONSTRAINT jbi_warehouses_pk
PRIMARY KEY (warehouse_id);
```

```
ALTER TABLE jbi_employees
ADD CONSTRAINT jbi_employees_employee_id_pk
PRIMARY KEY (employee_id);
```

```
ALTER TABLE jbi_employees
ADD CONSTRAINT jbi_employees_manager_id_fk
FOREIGN KEY (manager_id)
REFERENCES jbi_employees(employee_id);
```

```
ALTER TABLE jbi_employees
ADD CONSTRAINT jbi_employees_soc_sec_no_ck
CHECK (soc_sec_no between 11111111 and 99999999);
```

```
ALTER TABLE jbi_employees
ADD CONSTRAINT jbi_employees_soc_sec_no_uk
UNIQUE (soc_sec_no);
```

```
ALTER TABLE jbi_products
ADD CONSTRAINT jbi_products_product_id_pk
PRIMARY KEY (product_id);
```

```
ALTER TABLE jbi_customers
ADD CONSTRAINT jbi_customers_customer_id_pk
PRIMARY KEY (customer_id);
```

```
ALTER TABLE jbi_customers
ADD CONSTRAINT jbi_customers_email_uk
UNIQUE (email);
```

```
ALTER TABLE jbi_inventory
ADD CONSTRAINT jbi_inventory_pk
PRIMARY KEY (product_id, warehouse_id);
```

```
ALTER TABLE jbi_inventory
ADD CONSTRAINT jbi_inventory_product_id_fk
FOREIGN KEY (product_id)
REFERENCES jbi_products(product_id)
ON DELETE CASCADE
ON UPDATE RESTRICT;
```

```
ALTER TABLE jbi_inventory
ADD CONSTRAINT jbi_inventory_warehouse_id_fk
FOREIGN KEY (warehouse_id)
REFERENCES jbi_warehouses(warehouse_id)
ON DELETE CASCADE
ON UPDATE RESTRICT;
```

```
ALTER TABLE jbi_orders
ADD CONSTRAINT jbi_orders_pk
PRIMARY KEY (order_id);
```

```
ALTER TABLE jbi_orders
ADD CONSTRAINT jbi_orders_customer_id_fk
FOREIGN KEY (customer_id)
REFERENCES jbi_customers(customer_id)
ON DELETE CASCADE
ON UPDATE RESTRICT;
```

```
ALTER TABLE jbi_orders
ADD CONSTRAINT jbi_orders_employee_id_fk
FOREIGN KEY (employee_id)
REFERENCES jbi_employees(employee_id)
ON DELETE SET NULL;
```

```
ALTER TABLE jbi_products
ADD CONSTRAINT jbi_products_serial_number_uk
UNIQUE (serial_number);
```

```
ALTER TABLE jbi_products
ADD CONSTRAINT jbi_products_model_uk
UNIQUE (model);
```

```
ALTER TABLE jbi_order_lines
ADD CONSTRAINT jbi_order_lines_pk
PRIMARY KEY (order_id, product_id);
```

```
ALTER TABLE jbi_order_lines
ADD CONSTRAINT jbi_order_lines_order_id_fk
FOREIGN KEY (order_id)
REFERENCES jbi_orders(order_id)
ON DELETE CASCADE
ON UPDATE RESTRICT;
```

```
ALTER TABLE jbi_order_lines
ADD CONSTRAINT jbi_order_lines_product_id_fk
FOREIGN KEY (product_id)
REFERENCES jbi_products(product_id)
ON DELETE CASCADE
ON UPDATE RESTRICT;
```

```
ALTER TABLE jbi_regions
ADD CONSTRAINT jbi_regions_pk
PRIMARY KEY (region_id);
```

```
ALTER TABLE jbi_regions
ADD CONSTRAINT jbi_regions_region_name_uk
UNIQUE (region_name);
```

```
ALTER TABLE jbi_locations
ADD CONSTRAINT jbi_locations_pk
PRIMARY KEY (location_id);
```

```
ALTER TABLE jbi_locations
ADD CONSTRAINT jbi_locations_region_id_fk
FOREIGN KEY (region_id)
REFERENCES jbi_regions(region_id);
```

```
ALTER TABLE jbi_warehouses
ADD CONSTRAINT jbi_warehouses_location_id_fk
FOREIGN KEY (location_id)
REFERENCES jbi_locations(location_id);
```

```
ALTER TABLE jbi_warehouses
ADD CONSTRAINT jbi_warehouses_postal_code_uk
UNIQUE (postal_code);
```

Insert Statements

```
INSERT INTO jbi_regions (region_id, region_name)
VALUES (1, 'Southern Ontario'),
       (2, 'Eastern Ontario'),
       (3, 'Manitoba'),
       (4, 'Saskatchewan'),
       (5, 'Northern Alberta'),
       (6, 'Southern Alberta'),
       (7, 'British Columbia'),
       (8, 'Quebec'),
       (9, 'Maritimes');
```

```
INSERT INTO jbi_locations (location_id, city, region_id)
VALUES (1, 'Toronto', 1),
       (2, 'Ottawa', 2),
       (3, 'Winnipeg', 3),
       (4, 'Regina', 4),
       (5, '15820 87 Ave', 5),
       (6, 'Calgary', 6),
       (7, 'Vancouver', 7),
       (8, 'Montreal', 8),
       (9, 'Halifax', 9),
       (10, 'London', 1),
       (11, 'Hamilton', 1),
       (12, 'Kingston', 2);
```

```
INSERT INTO jbi_warehouses (warehouse_id, location_id, street_address,
postal_code)
VALUES (1, 1, '1060 Yonge Street', 'M4W 2L4'),
       (2, 2, '555 Mackenzie Ave', 'K1N 1K4'),
       (3, 3, '1555 Regent Ave', 'R2C 4J2'),
       (4, 4, '2720 College Ave', 'S4T 1T9'),
       (5, 5, '15820 87 Ave', 'T5R 5W9'),
       (6, 6, '601 8 Ave SW', 'T2P 0A7'),
       (7, 7, '635 Robson Street', 'V6B 5J3'),
       (8, 8, '433 Mayor Street', 'H3A 1N9'),
       (9, 9, '6324 Quinpool Road', 'B3L 1A5'),
       (10, 10, '160 Dundas Street W', 'N6A 1G1'),
       (11, 11, '41 King William Street', 'L8R 1A2'),
       (12, 12, '285 Princess Street', 'K7L 1B4');
```



```

INSERT INTO jbi_customers (customer_id, first_name, last_name, email,
phone)
VALUES (1, 'Jon', 'Mooney', 'jmooney@gmail.com', NULL),
      (2, 'Shana', 'Andrade', 'sandrade@gmail.com', '5196841143'),
      (3, 'Charli', 'Lowry', 'clowry@gmail.com', '5196846691'),
      (4, 'Murray', 'Briggs', 'mbriggs@gmail.com', '5196840877'),
      (5, 'Dominick', 'Mcdonnell', 'dmcdonnell@gmail.com',
'5196847770'),
      (6, 'Zephaniah', 'Contreras', 'zcontreras@gmail.com',
'5196844444'),
      (7, 'Neal', 'Myers', 'nmyers@gmail.com', '5196841234'),
      (8, 'Jess', 'Rawlings', 'jrawlings@gmail.com', '5196849941'),
      (9, 'Loren', 'Simons', 'lsimons@gmail.com', '5196843312'),
      (10, 'Keenan', 'Bowden', 'kbowden@gmail.com', '5196840008');

```

```

INSERT INTO jbi_employees (employee_id, first_name, last_name,
soc_sec_no, hire_date, salary, manager_id)
VALUES (20, 'Mariyah', 'Espinosa', 777645321, '2012-05-23', 95000.00,
null),
      (21, 'Brady', 'Curran', 777334955, '2014-12-01', 77500.00, 20),
      (22, 'James', 'Folger', 777461813, '2015-03-16', 71000.00, 20),
      (23, 'Orson', 'Kelly', 777551308, '2015-03-17', 64200.00, 20),
      (24, 'Ryan', 'Huffman', 777760458, '2016-07-08', 91000.00,
null),
      (25, 'Jacque', 'Iles', 777761333, '2014-09-22', 77000.00, 24),
      (26, 'Carmen', 'Sutton', 777053156, '2016-01-21', 81000.00,
24),
      (27, 'Libbi', 'Glass', 777996314, '2018-12-16', 79000.00, 24),
      (28, 'Angela', 'Corbett', 777778621, '2018-05-24', 68500.00,
20);

```

```

INSERT INTO jbi_products (product_id, model, brand, brew_type,
serial_number, item_description, rating, colour, cup_capacity,
programmable, quick_prep, small_space, price)
VALUES (9001022, 'CE201', 'Ninja', 'Drip', 'vNsXHbfLte', '12-Cup
Coffee Maker', 4.50, 'Black', 12, 1, 0, 0, 129.19),
(9001046, 'KF715BK', 'Braun', 'Drip', 'BgfUNeBCVo', 'Brew Sense
Drip Maker', 4.25, 'Black', 12, 1, 0, 0, 99.99),
(9001033, '611247373590', 'Keurig', 'Single Serve',
'2zVDJmGLvT', 'K-Mini Single Serv', 3.75, 'Black', 1, 1, 0, 0, 68.00),
(9001014, 'KU5055746', 'Keurig', 'Single Serve', 'k45n5HFXbP',
'K200 Brewing System', 4.00, 'Black', 1, 1, 0, 0, 149.99),
(9001003, 'BVSBTF6', 'Sunbeam', 'Drip', '3FvqtA9FWP', '5-Cup
Coffee Maker - Black', 4.75, 'Black', 5, 1, 0, 0, 18.98),
(9001019, 'DCC3400', 'Cuisinart', 'Drip', 'icLcN6uWGN', '12-Cup
Programmable', 4.25, 'Black', 12, 1, 0, 0, 145.27),
(9001023, 'JSUMZ0CV', 'Ninja', 'Drip', 'njx8hjcrFJ', 'Multi-
Serve Coffee Bar', 3.5, 'Black', 12, 1, 0, 0, 185.99),
(9001011, 'XNAX1SBR', 'Hamilton Beach', 'Drip', 's3tp2tXLbc',
'12-Cup BrewStation', 4.00, 'Black', 12, 1, 0, 0, 159.88),
(9001045, 'TJ3FA1VC', 'Keurig', 'Single Serve', 'SzVmLw5CPd',
'FlexBrew 2-Way Coffee Maker', 4.50, 'Black', 1, 1, 0, 0, 125.10),
(9001037, 'RDBEFT5B', 'Braun', 'Drip', 'yAsW8RhDmb',
'Multiserve Programmable', 4.25, 'Black', 12, 1, 0, 0, 89.99),
(9001047, 'CM1105BC', 'Black+Decker', 'Drip', '3YH7294Y', '12
Cup Programmable', 4.5, 'Black', 1, 1, 1, 0, 25.99),
(9001048, 'B07W56', 'Keurig', 'Single Serve', '5rkWi5Ty', 'K-
Duo Single Serve', 4.5, 'Black', 1, 0, 1, 1, 149.99),
(9001049, 'B00942', 'Sunbeam', 'Drip', 'AswH2rG6',
'Programmable Coffeemaker', 4, 'Silver', 12, 1, 0, 1, 39.98),
(9001050, 'DCC1200C', 'Cuisinart', 'Drip', 'DeK3iUyB', 'Brew
Central 12-Cup Programmable', 4.5, 'Silver', 12, 1, 0, 1, 89.88),
(9001051, 'B071RYF', 'Black+Decker', 'Drip', 'E8LuNqD9',
'Thermal Coffee Maker 12-Cup', 4.5, 'Silver', 12, 1, 0, 1, 64.99),
(9001052, 'KF7150BK', 'Braun', 'Drip', 'EkCRNQuW', 'Brew Sense
Drip Coffee Maker', 4.5, 'Black', 12, 1, 0, 1, 79.99),
(9001053, 'BVSBTF7', 'Sunbeam', 'Drip', 'EmkQvsTE', '5-Cup
Switch Coffee Maker - White', 4, 'White', 5, 0, 0, 1, 18.98),
(9001054, 'DCC3200C', 'Cuisinart', 'Drip', 'F9prtZUJ', '14 Cup
Programmable', 4.5, 'Silver', 14, 1, 0, 1, 89.99),
(9001055, 'KF7170', 'Braun', 'Drip', 'J7G4qNjg', 'BrewSense
Drip Coffee Maker', 4.5, 'Silver', 12, 1, 0, 1, 99.99),
(9001056, '49980A', 'Hamilton Beach', 'Single Serve',
'JTxVLCso', 'Single Serve Full Pot Coffee Maker', 4.5, 'Silver', 12,
1, 0, 1, 79.60),

```

(9001057, 'CM400C', 'Ninja', 'Single Serve', 'Jky8vF9U', 'Specialty Coffee Maker with Frother', 4, 'Silver', 6, 0, 1, 1, 211.63),

(9001058, 'KF6050WH', 'Braun', 'Drip', 'JyEagYUX', 'BrewSense Drip Coffee Maker', 4, 'Silver', 12, 1, 0, 1, 64.99),

(9001059, 'CM5000B', 'Black+Decker', 'Drip', 'KS5TQRXv', 'Mill & Brew Coffee Maker', 4, 'Black', 12, 1, 0, 1, 69.99),

(9001060, 'DGB-900BCC', 'Cuisinart', 'Drip', 'KqqMEN2y', 'Automatic Burr Grind & Brew Thermal', 4, 'Silver', 12, 1, 0, 1, 185.00),

(9001061, 'DCC-450BKC', 'Cuisinart', 'Drip', 'M5oZcb5S', '4-Cup Coffeemaker', 4.5, 'Black', 4, 0, 0, 1, 39.98),

(9001062, 'SS15C', 'Cuisinart', 'Drip', 'N8UGTtMp', '12-Cup Programmable', 4.5, 'Silver', 12, 1, 1, 1, 179.99),

(9001063, 'B07G14', 'Keurig', 'Single Server', 'PKKXa7FQ', 'K-Mini Coffee Maker', 4, 'Black', 1, 0, 1, 1, 59.99),

(9001064, 'CHW12C', 'Cuisinart', 'Drip', 'Piqn9CTA', '12-Cup Programmable', 4, 'Black', 12, 1, 0, 1, 107.62),

(9001065, 'B0768NNG', 'Hamilton Beach', 'Drip', 'QqcoNwWE', 'FlexBrew Coffee Maker', 4.5, 'Black', 12, 1, 1, 1, 115.73),

(9001066, '47380F', 'Hamilton Beach', 'Drip', 'SLL46Qsa', '10-Cup Brew Station', 4, 'Black', 10, 1, 0, 1, 42.98),

(9001067, 'CM4200SC', 'Black+Decker', 'Drip', 'TjwDhEU6', 'Select-A-Size Easy Dial', 4, 'Black', 12, 1, 1, 1, 39.00),

(9001068, '47380C', 'Hamilton Beach', 'Drip', 'VLy8NjBR', '10-Cup Brew Station', 4, 'Black', 10, 1, 0, 1, 42.98),

(9001069, 'B07B8SR', 'Keurig', 'Single Serve', 'WQW9waAb', 'K-Elite Coffee Maker Brushed Slate', 4.5, 'Brushed Slate', 1, 0, 1, 1, 168.99),

(9001070, 'KF7370SI', 'Braun', 'Drip', 'WpCqkhqs', 'BrewSense Touch Screen', 4.5, 'Silver', 12, 1, 0, 1, 99.99),

(9001071, 'DTC-975BKN', 'Cuisinart', 'Drip', 'XtyBgWE9', '12-Cup Thermal Coffee Maker', 4.5, 'Silver', 12, 1, 0, 1, 118.23),

(9001072, 'B0768NNO', 'Hamilton Beach', 'Drip', 'Y6K3n5qy', 'FlexBrew Coffee Maker', 4.5, 'Black', 12, 1, 0, 1, 89.99),

(9001073, 'B07B8SQXLK', 'Keurig', 'Single Serve', 'Yr3tBf7C', 'K-Select Coffee Maker', 4.5, 'Black', 1, 0, 1, 1, 102.00),

(9001074, 'B071FQS4TM', 'Keurig', 'Single Serve', 'ZhKKggao', 'K-Compact Coffee Maker', 4.5, 'Black', 1, 0, 1, 1, 98.00),

(9001075, 'DGB900BCC', 'Cuisinart', 'Drip', 'aEdC7ZeE', 'Automatic Burr Grind & Brew', 4, 'Silver', 12, 1, 0, 1, 192.93),

(9001076, 'KF6050BK', 'Braun', 'Drip', 'au9YeAfR', 'Brewsense Drip Coffee Maker', 4, 'Black', 12, 1, 0, 1, 95.49);

```
INSERT INTO jbi_orders (order_id, customer_id, order_date,  
employee_id)
```

```
VALUES (28915, 5, '2017-09-26', 22),  
        (28916, 1, '2017-07-21', 22),  
        (28917, 2, '2018-02-11', 22),  
        (28921, 3, '2018-01-05', 21),  
        (28922, 4, '2018-08-10', 22),  
        (28923, 5, '2018-10-11', 21),  
        (28924, 2, '2018-09-29', 22),  
        (28925, 1, '2018-05-30', 21),  
        (28926, 2, '2018-04-21', 20),  
        (28927, 6, '2018-02-13', 25),  
        (28928, 7, '2018-03-11', 24),  
        (28929, 8, '2019-03-13', 23),  
        (28930, 9, '2019-03-27', 26),  
        (28931, 10, '2019-04-05', 27),  
        (28932, 9, '2019-04-18', 28),  
        (28933, 7, '2019-05-22', 27),  
        (28934, 4, '2019-06-30', 23),  
        (28935, 2, '2019-07-10', 24),  
        (28936, 1, '2019-08-15', 26),  
        (28937, 3, '2019-08-22', 24),  
        (28938, 4, '2019-08-24', 27),  
        (28939, 3, '2019-08-29', 26),  
        (28940, 7, '2019-09-02', 25),  
        (28941, 8, '2019-09-13', 28),  
        (28942, 2, '2019-10-20', 21);
```

```
INSERT INTO jbi_order_lines (order_id, product_id, quantity, price)
VALUES (28915, 9001046, 1, 99.99),
      (28915, 9001061, 1, 39.98),
      (28916, 9001055, 1, 99.99),
      (28916, 9001023, 1, 185.99),
      (28917, 9001037, 2, 89.99),
      (28917, 9001052, 2, 79.99),
      (28921, 9001045, 1, 125.10),
      (28921, 9001064, 1, 107.62),
      (28922, 9001022, 3, 129.19),
      (28922, 9001074, 3, 98.00),
      (28923, 9001014, 2, 149.99),
      (28923, 9001066, 2, 42.98),
      (28924, 9001022, 1, 129.19),
      (28924, 9001058, 1, 64.99),
      (28925, 9001019, 2, 145.27),
      (28925, 9001049, 2, 39.98),
      (28926, 9001045, 1, 125.10),
      (28926, 9001075, 1, 192.93),
      (28927, 9001003, 4, 18.98),
      (28927, 9001019, 4, 145.27),
      (28928, 9001046, 3, 99.99),
      (28928, 9001056, 3, 79.60),
      (28929, 9001033, 2, 68.00),
      (28929, 9001050, 2, 89.88),
      (28930, 9001011, 1, 159.88),
      (28930, 9001054, 1, 89.99),
      (28931, 9001014, 1, 149.99),
      (28931, 9001048, 1, 149.99),
      (28932, 9001033, 3, 68.00),
      (28932, 9001053, 3, 18.98),
      (28933, 9001046, 2, 99.99),
      (28933, 9001003, 2, 18.98),
      (28934, 9001045, 1, 125.10),
      (28934, 9001014, 1, 149.99),
      (28935, 9001019, 2, 145.27),
      (28935, 9001051, 2, 64.99),
      (28936, 9001003, 3, 18.98),
      (28936, 9001011, 2, 159.88),
      (28937, 9001003, 1, 18.98),
      (28937, 9001037, 4, 89.99),
      (28938, 9001003, 2, 18.98),
      (28938, 9001056, 4, 79.60),
      (28939, 9001003, 2, 18.98),
      (28939, 9001060, 3, 185.00),
      (28940, 9001003, 2, 18.98),
```

```
(28940, 9001062, 3, 179.99),  
(28941, 9001003, 1, 18.98),  
(28941, 9001066, 3, 42.98),  
(28942, 9001003, 3, 18.98),  
(28942, 9001073, 3, 102.00);
```

```
INSERT INTO jbi_inventory (product_id, warehouse_id, quantity)
```

```
VALUES (9001022, 1, 14),  
(9001046, 1, 5),  
(9001033, 1, 14),  
(9001014, 1, 18),  
(9001003, 1, 5),  
(9001019, 1, 14),  
(9001023, 1, 3),  
(9001011, 1, 18),  
(9001045, 1, 14),  
(9001037, 1, 5),  
(9001047, 1, 6),  
(9001048, 1, 1),  
(9001049, 1, 3),  
(9001050, 1, 4),  
(9001051, 1, 8),  
(9001052, 1, 9),  
(9001053, 1, 8),  
(9001054, 1, 11),  
(9001055, 1, 6),  
(9001056, 1, 7),  
(9001057, 1, 3),  
(9001058, 1, 3),  
(9001059, 1, 1),  
(9001060, 1, 8),  
(9001061, 1, 4),  
(9001062, 1, 6),  
(9001063, 1, 4),  
(9001064, 1, 11),  
(9001065, 1, 10),  
(9001066, 1, 0),  
(9001067, 1, 0),  
(9001068, 1, 0),  
(9001069, 1, 4),  
(9001070, 1, 3),  
(9001071, 1, 1),  
(9001072, 1, 6),  
(9001073, 1, 6),  
(9001074, 1, 4),
```

(9001075, 1, 2),
(9001076, 1, 2),
(9001022, 2, 14),
(9001046, 2, 5),
(9001033, 2, 14),
(9001014, 2, 18),
(9001003, 2, 5),
(9001019, 2, 14),
(9001023, 2, 3),
(9001011, 2, 18),
(9001045, 2, 14),
(9001037, 2, 5),
(9001047, 2, 6),
(9001048, 2, 1),
(9001049, 2, 3),
(9001050, 2, 4),
(9001051, 2, 8),
(9001052, 2, 9),
(9001053, 2, 8),
(9001054, 2, 11),
(9001055, 2, 6),
(9001056, 2, 7),
(9001057, 2, 3),
(9001058, 2, 3),
(9001059, 2, 1),
(9001060, 2, 8),
(9001061, 2, 4),
(9001062, 2, 6),
(9001063, 2, 4),
(9001064, 2, 11),
(9001065, 2, 10),
(9001066, 2, 0),
(9001067, 2, 0),
(9001068, 2, 0),
(9001069, 2, 4),
(9001070, 2, 3),
(9001071, 2, 1),
(9001072, 2, 6),
(9001073, 2, 6),
(9001074, 2, 4),
(9001075, 2, 2),
(9001076, 2, 2),
(9001022, 3, 14),
(9001046, 3, 5),
(9001033, 3, 14),
(9001014, 3, 18),

(9001003, 3, 5),
(9001019, 3, 14),
(9001023, 3, 3),
(9001011, 3, 18),
(9001045, 3, 14),
(9001037, 3, 5),
(9001047, 3, 6),
(9001048, 3, 1),
(9001049, 3, 3),
(9001050, 3, 4),
(9001051, 3, 8),
(9001052, 3, 9),
(9001053, 3, 8),
(9001054, 3, 11),
(9001055, 3, 6),
(9001056, 3, 7),
(9001057, 3, 3),
(9001058, 3, 3),
(9001059, 3, 1),
(9001060, 3, 8),
(9001061, 3, 4),
(9001062, 3, 6),
(9001063, 3, 4),
(9001064, 3, 11),
(9001065, 3, 10),
(9001066, 3, 0),
(9001067, 3, 0),
(9001068, 3, 0),
(9001069, 3, 4),
(9001070, 3, 3),
(9001071, 3, 1),
(9001072, 3, 6),
(9001073, 3, 6),
(9001074, 3, 4),
(9001075, 3, 2),
(9001076, 3, 2),
(9001022, 4, 14),
(9001046, 4, 5),
(9001033, 4, 14),
(9001014, 4, 18),
(9001003, 4, 5),
(9001019, 4, 14),
(9001023, 4, 3),
(9001011, 4, 18),
(9001045, 4, 14),
(9001037, 4, 5),

(9001047, 4, 6),
(9001048, 4, 1),
(9001049, 4, 3),
(9001050, 4, 4),
(9001051, 4, 8),
(9001052, 4, 9),
(9001053, 4, 8),
(9001054, 4, 11),
(9001055, 4, 6),
(9001056, 4, 7),
(9001057, 4, 3),
(9001058, 4, 3),
(9001059, 4, 1),
(9001060, 4, 8),
(9001061, 4, 4),
(9001062, 4, 6),
(9001063, 4, 4),
(9001064, 4, 11),
(9001065, 4, 10),
(9001066, 4, 0),
(9001067, 4, 0),
(9001068, 4, 0),
(9001069, 4, 4),
(9001070, 4, 3),
(9001071, 4, 1),
(9001072, 4, 6),
(9001073, 4, 6),
(9001074, 4, 4),
(9001075, 4, 2),
(9001076, 4, 2),
(9001022, 5, 14),
(9001046, 5, 5),
(9001033, 5, 14),
(9001014, 5, 18),
(9001003, 5, 5),
(9001019, 5, 14),
(9001023, 5, 3),
(9001011, 5, 18),
(9001045, 5, 14),
(9001037, 5, 5),
(9001047, 5, 6),
(9001048, 5, 1),
(9001049, 5, 3),
(9001050, 5, 4),
(9001051, 5, 8),
(9001052, 5, 9),

(9001053, 5, 8),
(9001054, 5, 11),
(9001055, 5, 6),
(9001056, 5, 7),
(9001057, 5, 3),
(9001058, 5, 3),
(9001059, 5, 1),
(9001060, 5, 8),
(9001061, 5, 4),
(9001062, 5, 6),
(9001063, 5, 4),
(9001064, 5, 11),
(9001065, 5, 10),
(9001066, 5, 0),
(9001067, 5, 0),
(9001068, 5, 0),
(9001069, 5, 4),
(9001070, 5, 3),
(9001071, 5, 1),
(9001072, 5, 6),
(9001073, 5, 6),
(9001074, 5, 4),
(9001075, 5, 2),
(9001076, 5, 2),
(9001022, 6, 14),
(9001046, 6, 5),
(9001033, 6, 14),
(9001014, 6, 18),
(9001003, 6, 5),
(9001019, 6, 14),
(9001023, 6, 3),
(9001011, 6, 18),
(9001045, 6, 14),
(9001037, 6, 5),
(9001047, 6, 6),
(9001048, 6, 1),
(9001049, 6, 3),
(9001050, 6, 4),
(9001051, 6, 8),
(9001052, 6, 9),
(9001053, 6, 8),
(9001054, 6, 11),
(9001055, 6, 6),
(9001056, 6, 7),
(9001057, 6, 3),
(9001058, 6, 3),

```
(9001059, 6, 1),  
(9001060, 6, 8),  
(9001061, 6, 4),  
(9001062, 6, 6),  
(9001063, 6, 4),  
(9001064, 6, 11),  
(9001065, 6, 10),  
(9001066, 6, 0),  
(9001067, 6, 0),  
(9001068, 6, 0),  
(9001069, 6, 4),  
(9001070, 6, 3),  
(9001071, 6, 1),  
(9001072, 6, 6),  
(9001073, 6, 6),  
(9001074, 6, 4),  
(9001075, 6, 2),  
(9001076, 6, 2);
```

Constraint Testing

```
-- CONSTRAINT TESTING

-- Constraint Test 1
-- Description: Confirm primary key constraint on product_id column in
jbi_products table
-- Expected result: Insert fails with duplicate key error
-- Action
    INSERT INTO jbi_products VALUES (9001022, 'CE201', 'Ninja', 'Drip',
'vNsXHbFLte', '12-Cup Coffee Maker', 4.50, 'Black', 12, 1, 0, 0,
129.19);
-- Result:
-- [SQL0803] Duplicate key value specified.

-- Constraint Test 2
-- Description: Confirm primary key constraint on region_id column in
jbi_regions table
-- Expected result: Insert fails with duplicate key error
-- Action
    INSERT INTO jbi_regions VALUES (1, 'Southern Ontario');
-- Result:
-- [SQL0803] Duplicate key value specified.

-- Constraint Test 3
-- Description: Confirm primary key constraint on location_id column
in jbi_locations table
-- Expected result: Insert fails with duplicate key error
-- Action
    INSERT INTO jbi_locations VALUES (1, 'Toronto', 1);
-- Result:
-- [SQL0803] Duplicate key value specified.

-- Constraint Test 4
-- Description: Confirm primary key constraint on warehouse_id column
in jbi_warehouses table
-- Expected result: Insert fails with duplicate key error
-- Action
    INSERT INTO jbi_warehouses VALUES (1, 1, '1060 Yonge Street', 'M4W
2L4');
-- Result:
-- [SQL0803] Duplicate key value specified.
```

```

-- Constraint Test 5
-- Description: Confirm primary key constraint on customer_id column
in jbi_customers table
-- Expected result: Insert fails with duplicate key error
-- Action
INSERT INTO jbi_customers VALUES (1, 'Jon', 'Mooney',
'jmooney@gmail.com', NULL);
-- Result:
-- [SQL0803] Duplicate key value specified.

-- Constraint Test 6
-- Description: Confirm primary key constraint on employee_id column
in jbi_employees table
-- Expected result: Insert fails with duplicate key error
-- Action
INSERT INTO jbi_employees VALUES (20, 'Mariyah', 'Espinosa',
777645321, '2012-05-23', 95000.00, null);
-- Result:
-- [SQL0803] Duplicate key value specified.

-- Constraint Test 7
-- Description: Confirm primary key constraint on order_id column in
jbi_orders table
-- Expected result: Insert fails with duplicate key error
-- Action
INSERT INTO jbi_orders VALUES (28915, 5, '2019-09-26', 22);
-- Result:
-- [SQL0803] Duplicate key value specified.

-- Constraint Test 8
-- Description: Confirm primary key constraint on order_id column in
jbi_order_lines table
-- Expected result: Insert fails with duplicate key error
-- Action
INSERT INTO jbi_order_lines VALUES (28915, 9001046, 1, 99.99);
-- Result:
-- [SQL0803] Duplicate key value specified.

-- Constraint Test 9
-- Description: Confirm primary key constraint on product_id column in
jbi_inventory table
-- Expected result: Insert fails with duplicate key error
-- Action
INSERT INTO jbi_inventory VALUES (9001022, 1, 14);
-- Result:
-- [SQL0803] Duplicate key value specified.

```

```

-- Constraint Test 9
-- Description: Confirm foreign key constraint on manager_id column in
jbi_employees table
-- Expected result: Insert fails, operation not allowed
-- Action
    INSERT INTO jbi_employees VALUES (29, 'James', 'Folger', 999999998,
'2015-03-16', 71000.00, 33);
-- Result:
-- [SQL0530] Operation not allowed by referential constraint
JBI_EMPLOYEES_MANAGER_ID_FK in IBM7905.

-- Constraint Test 10
-- Description: Confirm check constraint on soc_sec_no column in
jbi_employees table
-- Expected result: Operation not allowed for check constraint
-- Action
    INSERT INTO jbi_employees VALUES (29, 'James', 'Folger',
1234567899, '2015-03-16', 71000.00, 20);
-- Result:
-- [SQL0545] INSERT, UPDATE, or MERGE not allowed by CHECK constraint.

-- Constraint Test 11
-- Description: Confirm check constraint on soc_sec_no column in
jbi_employees table
-- Expected result: Operation not allowed for check constraint
-- Action
    INSERT INTO jbi_employees VALUES (29, 'James', 'Folger',
1234567899, '2015-03-16', 71000.00, 20);
-- Result:
-- [SQL0545] INSERT, UPDATE, or MERGE not allowed by CHECK constraint.

-- Constraint Test 12
-- Description: Confirm unique constraint on soc_sec_no column in
jbi_employees table
-- Expected result: Insert fails with duplicate key error
-- Action
    INSERT INTO jbi_employees VALUES (45, 'James', 'Folger', 777334955,
'2015-03-16', 71000.00, 20);
-- Result:
-- [SQL0803] Duplicate key value specified.

```

```
-- Constraint Test 13
-- Description: Confirm unique constraint on email column in
jbi_customers table
-- Expected result: Insert fails with duplicate key error
-- Action
  INSERT INTO jbi_customers VALUES (16, 'Charlie', 'Lowryy',
'clowry@gmail.com', '5196846691');
-- Result:
-- [SQL0803] Duplicate key value specified.

-- Constraint Test 14
-- Description: Confirm unique constraint on email column in
jbi_customers table
-- Expected result: Insert fails with duplicate key error
-- Action
  INSERT INTO jbi_customers VALUES (16, 'Charlie', 'Lowryy',
'clowry@gmail.com', '5196846691');
-- Result:
-- [SQL0803] Duplicate key value specified.
```

Views

PRODV1

```
CREATE OR REPLACE VIEW prodv1 AS  
SELECT p.product_id, model, brand, item_description, SUM(quantity)  
quantity, price  
FROM jbi_inventory i  
JOIN JBI_PRODUCTS p ON p.PRODUCT_ID = i.PRODUCT_ID  
GROUP BY p.PRODUCT_ID, item_description, model, price, brand;
```

PRODV2

```
CREATE OR REPLACE VIEW prodv2 AS  
SELECT cust.first_name || ' ' || cust.last_name AS customer_name,  
TO_CHAR(ord.order_date, 'MM/DD/YY') AS order_date, ord.order_id,  
prod.product_id, prod.item_description, ordline.quantity,  
ordline.price  
FROM jbi_customers cust  
JOIN jbi_orders ord ON ord.CUSTOMER_ID = cust.CUSTOMER_ID  
JOIN jbi_order_lines ordline ON ordline.ORDER_ID = ord.ORDER_ID  
JOIN jbi_products prod ON prod.PRODUCT_ID = ordline.PRODUCT_ID;
```