2016-12-02

Database Design

FRESH HUMIDIFIER COMPANY

JINGBING HUANG
STUDENT# c0686674

Contents

Conceptual Design

Requirements Analysis

Descriptive Overview

Fresh Humidifier Company – Retailer

The Fresh Humidifier Company is a retailer that sells all kinds of humidifiers in three categories – household, commercial and industrial. The products are provided by different brand suppliers. Customers can buy humidifiers either in shop or on its own website. Delivery service is offered to meet customers' requirement.

The company holds and sells humidifiers products. Each humidifier is part of a product table that holds the product id, model, brand, category, color, width, depth, height, weight, power, applicable area, humidification capacity, manufacture date, description and item price.

Each customer is identified by a unique identifier called customer id. In addition, each customer's first name, last name, address and zip code is stored.

Each associate is identified by a unique identifier called associate id. In addition, each associate's name is stored.

Each order is identified by a unique identifier called order id. In addition, each order's order date is stored, along with associate id and customer id.

Sales Invoice

Fresh Humidifier Company

Customer Sales Invoice

767 Lambton Court,

Sarnia, N7S 6B8

Store: 001

Customer ID: 1001

Customer Name: Karen Smith

Customer Address: 625 Humber Drive

Sarnia, ON N7S 4K5

Order ID: 10001

Order Date: 2016-10-05

Associate ID: 001

Associate Name: Lucy Chen

Model	Brand	Type	Color	Size	weight	Power	Applicable Area	Humidification Capacity	Manufacture Date	Description	Quantity	Item Price	Extended Price
A001	LG	industrial	silver	10.5W x 11D x 29"H	6kg	40W	500sq.ft	500ml/h	2016-09-27	Latest Model	1	459.00	459.00
K103	KPGS	household	white	10.5W x 11D x 29"H	2kg	25W	115sq.ft	200ml/h	2016-09-27	Best seller	2	239.00	478.00
												Sub-total	937.00
												Tax	121.81
												Invoice Total	$1058.81

ER Diagram based on Requirements Analysis

Entities and Attributes

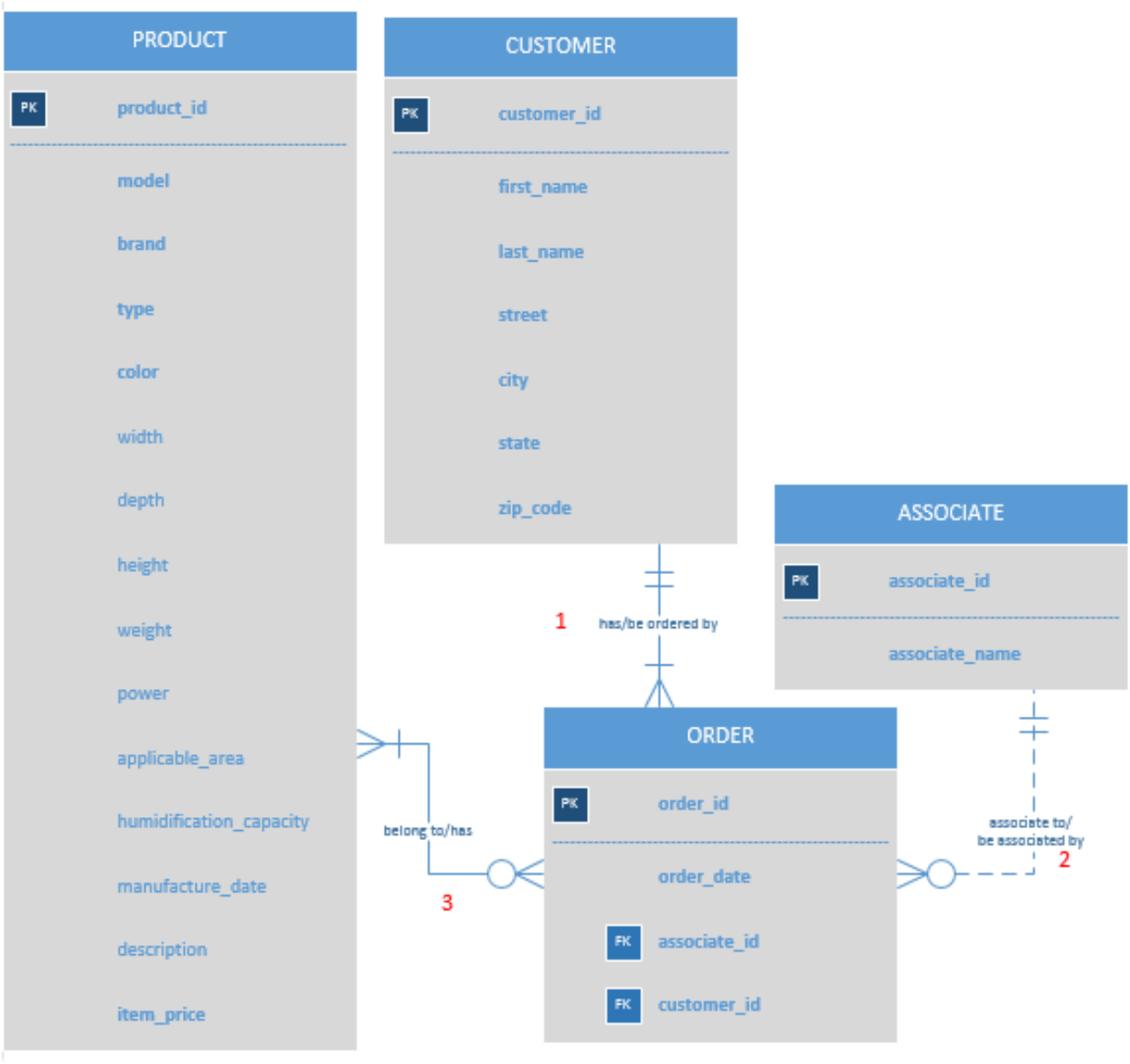
CUSTOMER		
ATTRIBUTE	DATA TYPE	LENGTH
Customer Id(UID)	INTEGER	5
First Name	VARCHAR	20
Last Name	VARCHAR	20
Street	VARCHAR	50
City	VARCHAR	20
Province	VARCHAR	20
Zip Code	VARCHAR	10

ASSOCIATE		
ATTRIBUTE	DATA TYPE	LENGTH
Associate Id(UID)	INTEGER	5
Associate Name	VARCHAR	30

ORDER		
ATTRIBUTE	DATA TYPE	LENGTH
Order Id	INTEGER	5
Order Date	DATE	10
Associate Id	INTEGER	5
Customer Id	INTEGER	5

PRODUCT		
ATTRIBUTE	DATA TYPE	LENGTH
Product Id	INTEGER	5
Model	VARCHAR	10
Brand	VARCHAR	20
Category	VARCHAR	10
Color	VARCHAR	10
Width	INTEGER	3
Depth	INTEGER	3
Height	INTEGER	3
Weight	INTEGER	2
Power	INTEGER	2
Applicable Area	INTEGER	3
Humidification Capacity	INTEGER	3
Manufacture Date	DATE	10
Description	VARCHAR	50
Item Price	DECIMAL	7,2

ER Diagram


PRODUCT
PK
product_id
model
brand
type
color
width
depth
height
weight
power
applicable_area
humidification_capacity
manufacture_date
description
item_price
CUSTOMER
PK
customer_id
first_name
last_name
street
city
state
zip_code
ASSOCIATE
PK
associate_id
associate_name
1
has/be ordered by
ORDER
PK
order_id
order_date
FK
associate_id
FK
customer_id
belong to/has
3
associate to/
be associated by
2

ERD resolving many-to-many relationships

Entities and Attributes

CUSTOMER		
ATTRIBUTE	DATA TYPE	LENGTH
Customer Id(UID)	INTEGER	5
First Name	VARCHAR	20
Last Name	VARCHAR	20
Street	VARCHAR	50
City	VARCHAR	20
Province	VARCHAR	20
Zip Code	VARCHAR	10

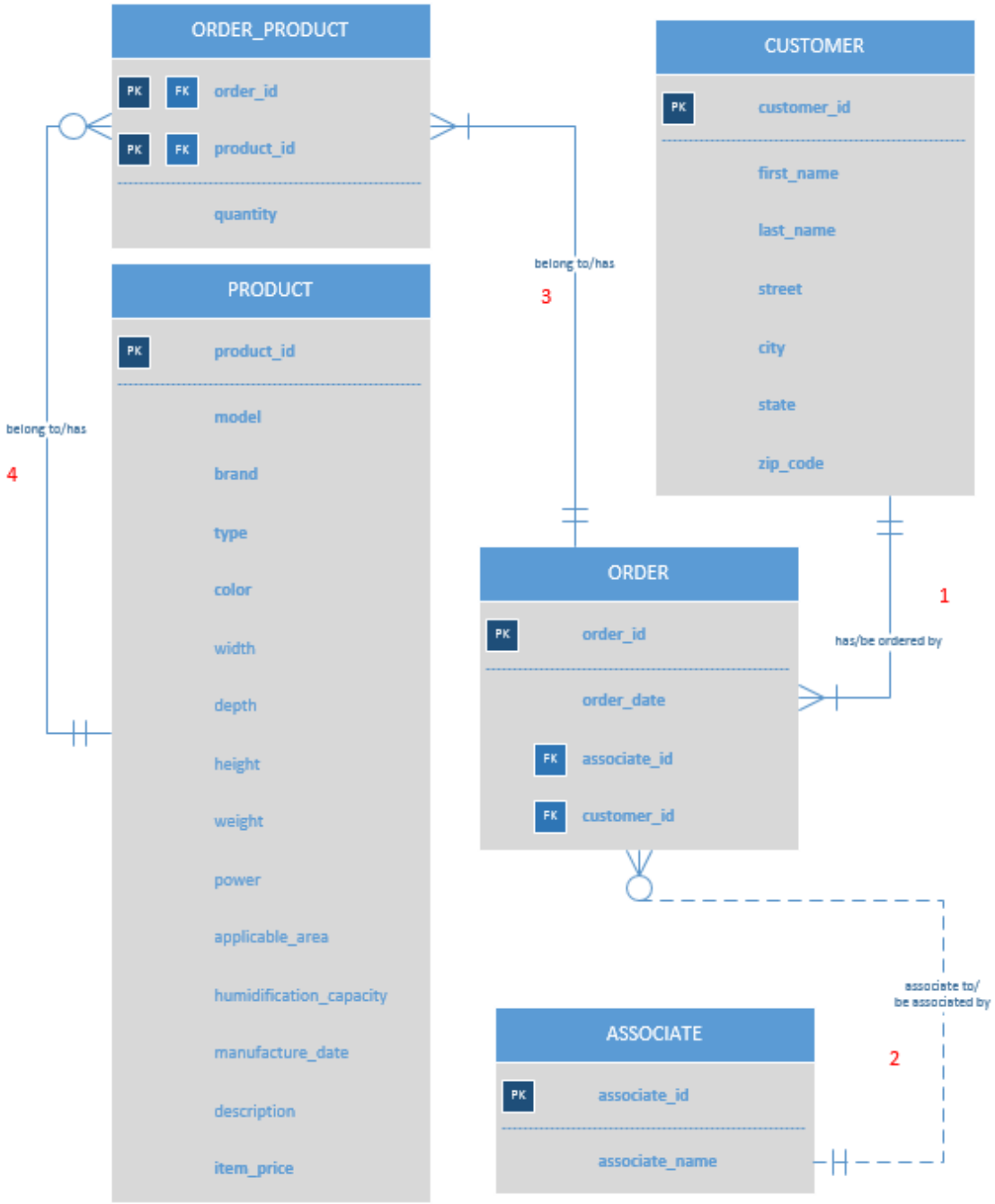
ASSOCIATE		
ATTRIBUTE	DATA TYPE	LENGTH
Associate Id(UID)	INTEGER	5
Associate Name	VARCHAR	30

ORDER		
ATTRIBUTE	DATA TYPE	LENGTH
Order Id	INTEGER	5
Order Date	DATE	10
Associate Id	INTEGER	5
Customer Id	INTEGER	5

PRODUCT		
ATTRIBUTE	DATA TYPE	LENGTH
Product Id	INTEGER	5
Model	VARCHAR	10
Brand	VARCHAR	20
Category	VARCHAR	10
Color	VARCHAR	10
Width	INTEGER	3
Depth	INTEGER	3
Height	INTEGER	3
Weight	INTEGER	2
Power	INTEGER	2
Applicable Area	INTEGER	3
Humidification Capacity	INTEGER	3
Manufacture Date	DATE	10
Description	VARCHAR	50
Item Price	DECIMAL	7,2

ORDER_PRODUCT		
ATTRIBUTE	DATA TYPE	LENGTH
Order Id	INTEGER	5
Product Id	INTEGER	5
Quantity	INTEGER	3

ER Diagram



Logical Data Model

Normalization and Relational Schema

CUSTOMER (**customer_id**, first_name, last_name, street, city, province, zip_code)

ORDER (**order_id**, order_date, *associate_id, customer_id*)

FK associate_id→ASSOCIATE

FK customer_id→CUSTOMER

ASSOCIATE (**associate_id**, associate_name)

PRODUCT (**product_id**, model, *brand*, *cg_id*, *color*, width, dep, height, weight, power, area, humidification_capacity, manufacture_date, item_price)

FK color→COLOR

FK cg_id→CATEGORY

FK brand→BRAND

ORDER_PRODUCT (***order_id***, ***product_id***, quantity)

FK order_id→ORDER

FK product_id→PRODUCT

COLOR (**color_id**, color_name)

CATEGORY (**category_id**, category_name)

BRAND (**brand_id**, brand_name)

CUSTOMER						
customer_id	first_name	last_name	street	city	province	zip_code
1001	Karen	Smith	625 Humber Drive	Sarnia	Ontario	N7S6B8
1002	Jimmy	Allen	439 Butterfly Court	Windsor	Ontario	N9K1E9
1003	Tom	Johnson	228 Louis Drive	Toronto	Ontario	M9W5L9
1004	Rachel	Johnson	228 Louis Drive	Toronto	Ontario	M9W5L9
1005	Helen	Jones	159 Campus Street	London	Ontario	N6P1V2

ASSOCIATE	
associate_id	associate_name
123	Lucy Chen
345	Jack Smith
567	Betty Liu

ORDER		
order_id	order_date	*associate_id*
10001	2016-10-05	123
10002	2016-10-12	345
10003	2016-10-17	567
10004	2016-10-20	123
10005	2016-10-25	567

PRODUCT													
product_id	Model	*brand*	*cg_id*	*color*	width	dep	height	weight	power	area	capacity	manufacture _date	item _price
1001	A001	111	3	11	10	11	29	2	25	115	200	2016-09-27	459
3103	K103	222	1	22	21	22	35	4	40	250	500	2016-08-22	239
2102	C102	333	2	33	15	16	28	3	28	150	250	2016-05-12	566
1003	A003	111	1	44	18	21	37	4	30	200	350	2016-01-15	312

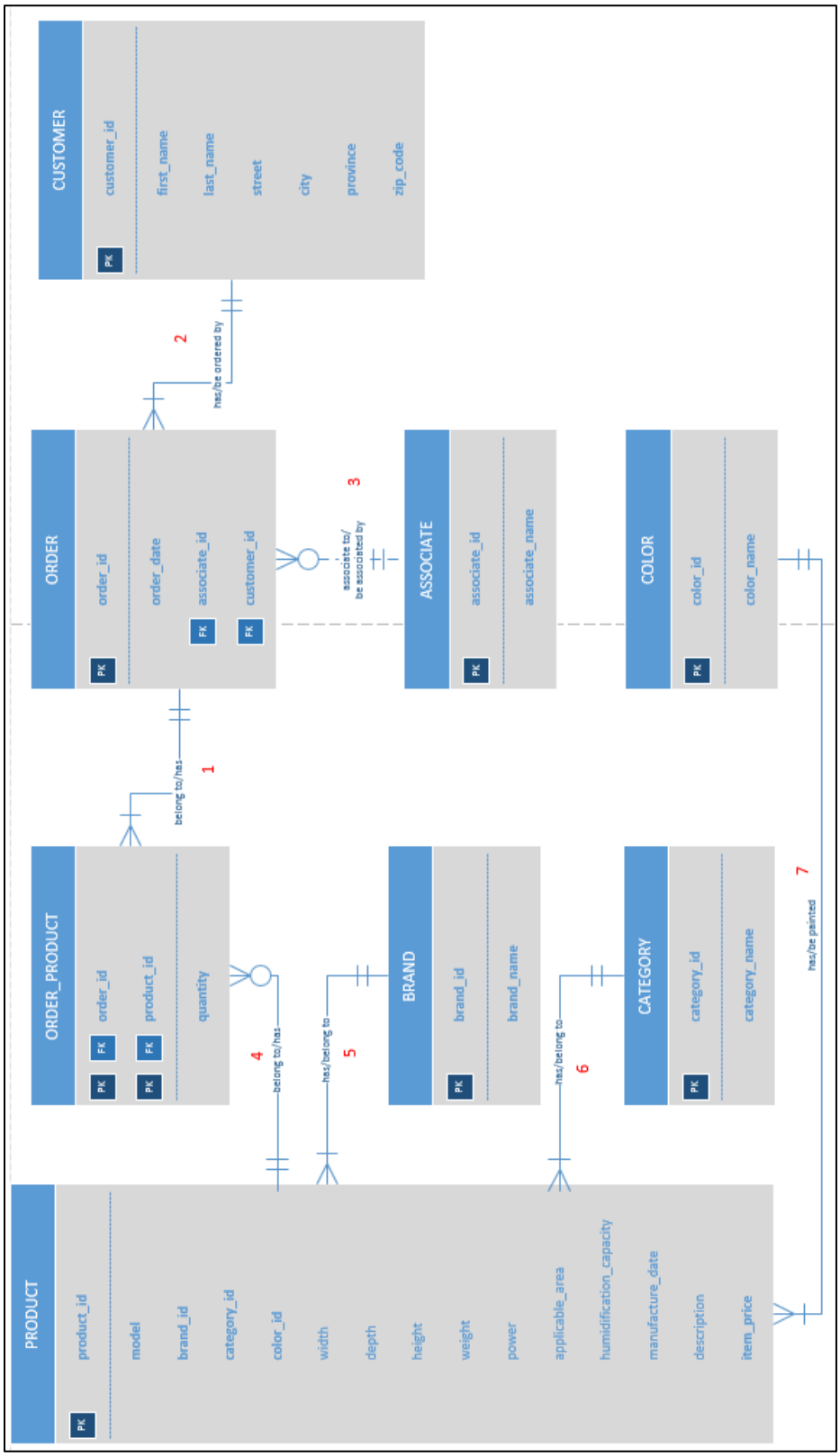
ORDER_PRODUCT		
order_id	*product_id*	quantity
10001	1001	1
10001	3103	2
10002	1001	2
10002	2102	1
10003	1003	2

COLOR	
color_id	color_name
11	Silver
22	White
33	Black
44	purple
55	Red

CATEGORY	
category_id	category_name
1	household
2	commercial
3	industrial

BRAND	
brand_id	brand_name
111	LG
222	KPGS
333	SIEMENS

Final ERD


CUSTOMER
customer_id
PK
first_name
last_name
street
city
province
zip_code
2
has/be ordered by
ORDER
order_id
PK
order_date
associate_id
FK
customer_id
FK
3
associate to/
be associated by
ASSOCIATE
associate_id
PK
associate_name
COLOR
color_id
PK
color_name
1
belong to/has
ORDER_PRODUCT
order_id
FK
PK
product_id
FK
PK
quantity
4
belong to/has
5
has/belong to
BRAND
brand_id
PK
brand_name
6
has/belong to
CATEGORY
category_id
PK
category_name
7
has/be painted
PRODUCT
product_id
PK
model
brand_id
category_id
color_id
width
depth
height
weight
power
applicable_area
humidification_capacity
manufacture_date
description
item_price

Relationships using ERD Language

CUSTOMER-ORDER

- Each CUSTOMER must have one or more ORDERs.
- Each ORDER must be ordered by one and only one CUSTOMER.

ASSOCIATE-ORDER

- Each ASSOCIATE may associate with zero, one or more ORDERs.
- Each ORDER must be associated by one and only one ASSOCIATE.

PRODUCT-BRAND

- Each PRODUCT must belong to one and only one BRAND.
- Each BRAND must have one or more PRODUCTs.

PRODUCT-CATEGORY

- Each PRODUCT must belong to one and only one CATEGORY.
- Each CATEGORY must have one or more PRODUCTs.

PRODUCT-COLOR

- Each PRODUCT must be painted in one and only one COLOR.
- Each COLOR must be painted to one or more PRODUCTs.

PRODUCT-ORDER_PRODUCT

- Each PRODUCT may belong to zero, one or more ORDER_PRODUCTs.
- Each ORDER_PRODUCT must have one and only one PRODUCT.

ORDER-ORDER_PRODUCT

- Each ORDER must belong to one or more ORDER_PRODUCTs.
- Each ORDER_PRODUCT must have one and only one ORDER.

Physical Database Design

Table Definitions

CUSTOMER		
ATTRIBUTE	DATA TYPE	LENGTH
customer_id(PK)	INTEGER	5
first_name	VARCHAR	20
last_name	VARCHAR	20
Street	VARCHAR	50
City	VARCHAR	20
province	VARCHAR	20
zip_code	VARCHAR	10

ASSOCIATE		
ATTRIBUTE	DATA TYPE	LENGTH
associate_id(PK)	INTEGER	5
associate_name	VARCHAR	20

ORDER		
ATTRIBUTE	DATA TYPE	LENGTH
order_id(PK)	INTEGER	5
order_date	DATE	10
associate_id(FK)	INTEGER	5
customer_id(FK)	INTEGER	5

ORDER_PRODUCT		
ATTRIBUTE	DATA TYPE	LENGTH
order_id(PK, FK)	INTEGER	5
product_id(PK, FK)	INTEGER	5
quantity	INTEGER	3

COLOR		
ATTRIBUTE	DATA TYPE	LENGTH
color_id(PK)	INTEGER	2
color_name(UK)	VARCHAR	20

PRODUCT		
ATTRIBUTE	DATA TYPE	LENGTH
product_id(PK)	INTEGER	5
Model(UK)	VARCHAR	10
brand_id(FK)	INTEGER	3
category_id(FK)	INTEGER	1
color_id(FK)	INTEGER	2
Width	INTEGER	3
Depth	INTEGER	3
Height	INTEGER	3
Weight	INTEGER	2
Power	INTEGER	2
applicable_area	INTEGER	3
humidification_capacity	INTEGER	3
manufacture_date	DATE	10
item_price	DECIMAL	7,2

CATEGORY		
ATTRIBUTE	DATA TYPE	LENGTH
category_id(PK)	INTEGER	1
category_name(UK)	VARCHAR	20

BRAND		
ATTRIBUTE	DATA TYPE	LENGTH
brand_id(PK)	INTEGER	3
brand_name(UK)	VARCHAR	20

Business Rules & Constraints

CUSTOMER

- Each customer is identified by a unique Customer ID
- Each customer has a first name
- Each customer has a last name
- Each customer has a street address
- Each customer live in a city
- Each customer live in a province
- Each customer has a zip code of the address

ASSOCIATE

- Each associate is identified by a unique Associate ID
- Each associate has an associate name between 100 and 999

ORDER

- Each order is identified by a unique Order ID
- Each order has an order date which must be 1980 or later and be current date by default
- Each order has an Associate ID, identified by the ASSOCIATE's Associate ID
- Each order has a Customer ID, identified by the CUSTOMER's Customer ID

PRODUCT

- Each product is identified by a unique Product ID between 1000 and 9999
- Each product has a model which must be unique
- Each product has a Brand ID, identified by BRAND's Brand ID
- Each product has a Category ID, identified by CATEGORY's Category ID
- Each product has a Color ID, identified by COLOR's Color ID
- Each product has a number of width, which must be greater than 0
- Each product has a number of depth, which must be greater than 0
- Each product has a number of height, which must be greater than 0
- Each product has a number of weight, which must be greater than 0

- Each product has a number of power between 15 and 40
- Each product has a number of applicable area, which must be greater than 0
- Each product has a number of humidification capacity, which must be greater than 0
- Each product has a manufacture date, which is the current date by default
- Each product has an item price, which must be greater than 0

ORDER_PRODUCT

- Each order_product is identified by a unique combination of Order ID and Product ID
- Each order_product has an Order ID, identified by ORDER's Order ID
- Each order_product has a Product ID, identified by PRODUCT's Product ID
- Each order_product has a quantity, which must be greater than 0

COLOR

- Each color is identified by a unique Color ID
- Each color has a unique color name

CATEGORY

- Each category is identified by a unique Category ID
- Each category has a unique category name, which must be included in "household", "commercial" and "industrial"

BRAND

- Each brand is identified by a unique Brand ID
- Each brand has a unique brand name

Database Table Creation

SQL Drop Statements

```
/* Drop tables */
DROP TABLE customers;
DROP TABLE associates;
DROP TABLE orders;
DROP TABLE products;
DROP TABLE order_products;
DROP TABLE colors;
DROP TABLE categories;
DROP TABLE brands;
```

SQL Create Statements

```
/* Create tables */
CREATE TABLE customers (
	customer_id		INTEGER		NOT NULL,
	first_name		VARCHAR(20)	NOT NULL,
	last_name		VARCHAR(20)	NOT NULL,
	street			VARCHAR(50)	NOT NULL,
	city			VARCHAR(20)	NOT NULL,
	province		VARCHAR(20)	NOT NULL,
	zip_code		VARCHAR(10)	NOT NULL
);

CREATE TABLE associates (
	associate_id		INTEGER		NOT NULL,
	associate_name		VARCHAR(20)	NOT NULL
);

CREATE TABLE orders (
	order_id		INTEGER		NOT NULL,
	order_date		DATE		NOT NULL,
	associate_id		INTEGER		NOT NULL,
	customer_id		INTEGER		NOT NULL
);

CREATE TABLE products (
	product_id			INTEGER		NOT NULL,
	model				VARCHAR(10)	NOT NULL,
	brand_id			INTEGER		NOT NULL,
	category_id			INTEGER		NOT NULL,
	color_id			INTEGER		NOT NULL,
	width				INTEGER,
	depth				INTEGER,
	height				INTEGER,
	weight				INTEGER,
	powers				INTEGER,
	applicable_area			INTEGER,
	humidification_capacity		INTEGER,
	manufacture_date		DATE,
```

```
      item_price                DECIMAL(7,2)        NOT NULL
);

CREATE TABLE order_products (
      order_id            INTEGER             NOT NULL,
      product_id          INTEGER             NOT NULL,
      quantity            INTEGER             NOT NULL
);

CREATE TABLE colors (
      color_id            INTEGER             NOT NULL,
      color_name          VARCHAR(20)         NOT NULL
);

CREATE TABLE categories (
      category_id         INTEGER             NOT NULL,
      category_name       VARCHAR(20)         NOT NULL
);

CREATE TABLE brands (
      brand_id            INTEGER             NOT NULL,
      brand_name          VARCHAR(20)         NOT NULL
);
```

SQL Constraint Statements

```
/* Primary key constraints */
ALTER TABLE customers
      ADD CONSTRAINT customers_customer_id_pk
      PRIMARY KEY ( customer_id );

ALTER TABLE associates
      ADD CONSTRAINT associates_associate_id_pk
      PRIMARY KEY ( associate_id );

ALTER TABLE orders
      ADD CONSTRAINT orders_order_id_pk
      PRIMARY KEY ( order_id );

ALTER TABLE products
      ADD CONSTRAINT products_product_id_pk
      PRIMARY KEY ( product_id );

ALTER TABLE order_products
      ADD CONSTRAINT order_products_pk
      PRIMARY KEY ( order_id, product_id );

ALTER TABLE colors
      ADD CONSTRAINT colors_pk
      PRIMARY KEY ( color_id );

ALTER TABLE categories
      ADD CONSTRAINT categories_pk
      PRIMARY KEY ( category_id );
ALTER TABLE brands
```

```
        ADD CONSTRAINT brands_pk
        PRIMARY KEY ( brand_id );

/* Unique key constraints */
ALTER TABLE products
        ADD CONSTRAINT products_model_uq
        UNIQUE ( model );

ALTER TABLE colors
        ADD CONSTRAINT colors_color_name_uq
        UNIQUE ( color_name );

ALTER TABLE categories
        ADD CONSTRAINT categories_category_name_uq
        UNIQUE ( category_name );

ALTER TABLE brands
        ADD CONSTRAINT brands_brand_name_uq
        UNIQUE ( brand_name );

/* Foreign key constraints */
ALTER TABLE orders
        ADD CONSTRAINT orders_associate_id_fk
        FOREIGN KEY ( associate_id )
        REFERENCES associates ( associate_id );

ALTER TABLE orders
        ADD CONSTRAINT orders_customer_id_fk
        FOREIGN KEY ( customer_id )
        REFERENCES customers ( customer_id );

ALTER TABLE products
        ADD CONSTRAINT products_brand_id_fk
        FOREIGN KEY ( brand_id )
        REFERENCES brands ( brand_id );

ALTER TABLE products
        ADD CONSTRAINT products_category_id_fk
        FOREIGN KEY ( category_id )
        REFERENCES categories ( category_id );

ALTER TABLE products
        ADD CONSTRAINT products_color_id_fk
        FOREIGN KEY ( color_id )
        REFERENCES colors ( color_id );

ALTER TABLE order_products
        ADD CONSTRAINT order_products_order_id_fk
        FOREIGN KEY (order_id)
        REFERENCES orders(order_id);

ALTER TABLE order_products
        ADD CONSTRAINT order_products_product_id_fk
        FOREIGN KEY ( product_id )
        REFERENCES products ( product_id );
```

```
/* Default constraints */
ALTER TABLE orders
       ALTER COLUMN order_date
       SET DEFAULT CURRENT_DATE;

ALTER TABLE products
       ALTER COLUMN manufacture_date
       SET DEFAULT CURRENT_DATE;

/* Check constraints */

ALTER TABLE associates
       ADD CONSTRAINT associates_associate_id_ck
       CHECK ( associate_id >= 100 AND associate_id <= 999 );

ALTER TABLE orders
       ADD CONSTRAINT orders_order_date_ck
       CHECK ( order_date >= '1980-01-01' );

ALTER TABLE products
       ADD CONSTRAINT products_product_id_ck
       CHECK ( product_id >= 1000 AND product_id <= 9999 );

ALTER TABLE products
       ADD CONSTRAINT products_width_ck
       CHECK ( width > 0 );

ALTER TABLE products
       ADD CONSTRAINT products_depth_ck
       CHECK ( depth > 0 );

ALTER TABLE products
       ADD CONSTRAINT products_height_ck
       CHECK ( height > 0 );

ALTER TABLE products
       ADD CONSTRAINT products_weight_ck
       CHECK ( weight > 0 );

ALTER TABLE products
       ADD CONSTRAINT products_powers_ck
       CHECK ( powers >= 15 AND powers <= 40 );

ALTER TABLE products
       ADD CONSTRAINT products_applicable_area_ck
       CHECK ( applicable_area > 0 );

ALTER TABLE products
       ADD CONSTRAINT products_humidification_capacity_ck
       CHECK ( humidification_capacity > 0 );

ALTER TABLE products
       ADD CONSTRAINT products_item_price_ck
       CHECK ( item_price > 0 );
```

```
ALTER TABLE order_products
      ADD CONSTRAINT invoices_quantity_ck
      CHECK ( quantity > 0 );

ALTER TABLE categories
      ADD CONSTRAINT categories_category_name_ck
      CHECK (category_name IN ('household', 'commercial', 'industrial'));
```

SQL Insert Statements

```
/* customers*/
INSERT INTO customers
VALUES (1001, 'Karen', 'Smith', '625 Humber Drive', 'Sarnia', 'Ontario', 'N7S6B8');

INSERT INTO customers
VALUES (1002, 'Jimmy', 'Allen', '439 Butterfly Court', 'Windsor', 'Ontario',
'N9K1E9');

INSERT INTO customers
VALUES (1003, 'Tom', 'Johnson', '228 Louis Drive', 'Toronto', 'Ontario', 'M9W5L9');

INSERT INTO customers
VALUES (1004, 'Rachel', 'Johnson', '228 Louis Drive', 'Toronto', 'Ontario',
'M9W5L9');

INSERT INTO customers
VALUES (1005, 'Helen', 'Jones', '159 Campus Street', 'London', 'Ontario', 'N6P1V2');

/* associates */
INSERT INTO associates
VALUES (111, 'Lucy Chen');

INSERT INTO associates
VALUES (222, 'Jack Smith');

INSERT INTO associates
VALUES (333, 'Mike Louis');

/* orders */
INSERT INTO orders
VALUES (1001, '2016-10-05', 111, 1001);

INSERT INTO orders
VALUES (1002, '2016-10-06', 222, 1002);

INSERT INTO orders
VALUES (1003, '2016-10-06', 333, 1003);

INSERT INTO orders
VALUES (1004, '2016-10-07', 333, 1004);

INSERT INTO orders
VALUES (1005, '2016-10-07', 222, 1005);

INSERT INTO orders
```

```
VALUES (1006, '2016-10-08', 111, 1002);

INSERT INTO orders
VALUES (1007, '2016-10-08', 222, 1003);

INSERT INTO orders
VALUES (1008, '2016-10-08', 333, 1004);

INSERT INTO orders
VALUES (1009, '2016-10-09', 111, 1005);

INSERT INTO orders
VALUES (1010, '2016-10-09', 333, 1001);

/* products */
INSERT INTO products
VALUES (1001, 'L001', 101, 1, 22, 10, 11, 29, 2, 25, 115, 200,'2016-09-27', 239);

INSERT INTO products
VALUES (1002, 'L002', 101, 3, 11, 21, 22, 35, 4, 40, 250, 500,'2016-08-22', 459);

INSERT INTO products
VALUES (1003, 'L003', 101, 2, 55, 18, 19, 30, 3, 35, 200, 350,'2015-10-02', 380);

INSERT INTO products
VALUES (2102, 'K102', 102, 3, 77, 25, 30, 35, 6, 45, 250, 450,'2014-01-14', 600);

INSERT INTO products
VALUES (2103, 'K103', 102, 1, 99, 12, 15, 20, 2, 20, 100, 100,'2015-02-22', 150);

INSERT INTO products
VALUES (2104, 'K104', 102, 1, 33, 15, 16, 18, 2, 20, 150, 150,'2016-07-12', 150);

INSERT INTO products
VALUES (2105, 'K105', 102, 2, 66, 24, 26, 30, 5, 30, 200, 250,'2016-11-01', 478);

INSERT INTO products
VALUES (3103, 'S103', 103, 2, 88, 28, 29, 30, 6, 40, 250, 300,'2015-05-06', 500);

INSERT INTO products
VALUES (3104, 'S104', 103, 1, 11, 18, 19, 20, 3, 25, 120, 225,'2016-08-14', 225);

INSERT INTO products
VALUES (3105, 'S105', 103, 3, 44, 30, 32, 35, 8, 45, 225, 450,'2014-12-11', 499);

/* order_products */
INSERT INTO order_products
VALUES (1001, 1002, 1);

INSERT INTO order_products
VALUES (1001, 3105, 2);

INSERT INTO order_products
VALUES (1002, 1003, 3);
```

```
INSERT INTO order_products
VALUES (1002, 2102, 4);

INSERT INTO order_products
VALUES (1003, 3104, 5);

INSERT INTO order_products
VALUES (1003, 1003, 6);

INSERT INTO order_products
VALUES (1004, 2104, 7);

INSERT INTO order_products
VALUES (1004, 3105, 8);

INSERT INTO order_products
VALUES (1005, 1002, 9);

INSERT INTO order_products
VALUES (1005, 3103, 10);

INSERT INTO order_products
VALUES (1006, 1002, 11);

INSERT INTO order_products
VALUES (1006, 3104, 12);

INSERT INTO order_products
VALUES (1007, 2105, 13);

INSERT INTO order_products
VALUES (1007, 1002, 14);

INSERT INTO order_products
VALUES (1008, 1003, 15);

INSERT INTO order_products
VALUES (1008, 3103, 16);

INSERT INTO order_products
VALUES (1009, 2103, 17);

INSERT INTO order_products
VALUES (1009, 1002, 18);

INSERT INTO order_products
VALUES (1010, 1003, 19);

INSERT INTO order_products
VALUES (1010, 2102, 20);
```

```
/* colors */
INSERT INTO colors
VALUES (11, 'black');

INSERT INTO colors
VALUES (22, 'white');

INSERT INTO colors
VALUES (33, 'blue');

INSERT INTO colors
VALUES (44, 'purple');

INSERT INTO colors
VALUES (55, 'silver');

INSERT INTO colors
VALUES (66, 'green');

INSERT INTO colors
VALUES (77, 'grey');

INSERT INTO colors
VALUES (88, 'yellow');

INSERT INTO colors
VALUES (99, 'red');

/* categories */
INSERT INTO categories
VALUES (1, 'household');

INSERT INTO categories
VALUES (2, 'commercial');

INSERT INTO categories
VALUES (3, 'industrial');


/* brands */
INSERT INTO brands
VALUES (101, 'LG');

INSERT INTO brands
VALUES (102, 'KPGS');

INSERT INTO brands
VALUES (103, 'SIEMENS');
```

SQL Select Statements

```
/* Check constraints */
SELECT * FROM customers;
SELECT * FROM associates;
SELECT * FROM orders;
SELECT * FROM products;
SELECT * FROM order_products;
SELECT * FROM colors;
SELECT * FROM categories;
SELECT * FROM brands;
```

SQL Constraint Tests

Customers Table

```
DROP TABLE customers;

CREATE TABLE customers (
	customer_id		INTEGER			NOT NULL,
	first_name		VARCHAR(20)		NOT NULL,
	last_name		VARCHAR(20)		NOT NULL,
	street			VARCHAR(50)		NOT NULL,
	city			VARCHAR(20)		NOT NULL,
	province		VARCHAR(20)		NOT NULL,
	zip_code		VARCHAR(10)		NOT NULL
);

/* primary key */
ALTER TABLE customers
	ADD CONSTRAINT customers_customer_id_pk
	PRIMARY KEY ( customer_id );

Working statement
INSERT INTO customers
VALUES (1001, 'Karen', 'Smith', '625 Humber Drive', 'Sarnia', 'Ontario', 'N7S6B8');

INSERT INTO customers
VALUES (1002, 'Jimmy', 'Allen', '439 Butterfly Court', 'Windsor', 'Ontario',
'N9K1E9');

Failing statement
INSERT INTO customers
VALUES (1001, 'Karen', 'Smith', '625 Humber Drive', 'Sarnia', 'Ontario', 'N7S6B8');

INSERT INTO customers
VALUES (1002, 'Jimmy', 'Allen', '439 Butterfly Court', 'Windsor', 'Ontario',
'N9K1E9');

[SQL0803] Duplicate key value specified.

Insert statement will fail to run as a duplicate customer_id was input.
--------------------------------------------------------------------------------
```

Associates Table

```
DROP TABLE associates;

CREATE TABLE associates (
      associate_id        INTEGER             NOT NULL,
      associate_name      VARCHAR(20)         NOT NULL
);

/* primary key */
ALTER TABLE associates
      ADD CONSTRAINT associates_associate_id_pk
      PRIMARY KEY ( associate_id );

Working statement
INSERT INTO associates
VALUES (111, 'Lucy Chen');

INSERT INTO associates
VALUES (222, 'Jack Smith');

Failing statement
INSERT INTO associates
VALUES (111, 'Lucy Chen');

INSERT INTO associates
VALUES (222, 'Jack Smith');

[SQL0803] Duplicate key value specified.

Insert statement will fail to run as a duplicate associate_id was input.
--------------------------------------------------------------------------------

/* check constraint */
ALTER TABLE associates
      ADD CONSTRAINT associates_associate_id_ck
      CHECK ( associate_id >= 100 AND associate_id <= 999 );

Working statement
INSERT INTO associates
VALUES (444, 'Amy');

Failing statement
INSERT INTO associates
VALUES (4444, 'Amy');

[SQL0545] INSERT, UPDATE, or MERGE not allowed by CHECK constraint.

Insert statement will fail to run as each associate_id must between 100 and 999.
--------------------------------------------------------------------------------
```

Orders Table

```
DROP TABLE orders;

CREATE TABLE orders (
       order_id            INTEGER             NOT NULL,
       order_date          DATE                NOT NULL,
       associate_id        INTEGER             NOT NULL,
       customer_id         INTEGER             NOT NULL
);

/* primary key */
ALTER TABLE orders
       ADD CONSTRAINT orders_order_id_pk
       PRIMARY KEY ( order_id );

Working statement
INSERT INTO orders
VALUES (1001, '2016-10-05', 111, 1001);

INSERT INTO orders
VALUES (1002, '2016-10-06', 222, 1002);

Failing statement
INSERT INTO orders
VALUES (1001, '2016-10-05', 111, 1001);

INSERT INTO orders
VALUES (1002, '2016-10-06', 222, 1002);

[SQL0803] Duplicate key value specified.

Insert statement will fail to run as a duplicate order_id was input.
--------------------------------------------------------------------------------

/* Foreign key constraints */
ALTER TABLE orders
       ADD CONSTRAINT orders_associate_id_fk
       FOREIGN KEY ( associate_id )
       REFERENCES associates ( associate_id );

Working statement
INSERT INTO orders
VALUES (2222, '2016-10-06', 333, 1002);

Failing statement
INSERT INTO orders
VALUES (3333, '2016-10-06', 444, 1002);

[SQL0530] Operation not allowed by referential constraint ORDERS_ASSOCIATE_ID_FK in
IBM20.

Insert statement will fail to run as each entry in the orders (associate_id) must
have an entry that is also a part of the associates (associate_id).
```

```
--------------------------------------------------------------------------------
ALTER TABLE orders
       ADD CONSTRAINT orders_customer_id_fk
       FOREIGN KEY ( customer_id )
       REFERENCES customers ( customer_id );

Working statement
INSERT INTO orders
VALUES (4444, '2016-10-06', 222, 1003);

Failing statement
INSERT INTO orders
VALUES (5555, '2016-10-06', 222, 9999);

[SQL0530] Operation not allowed by referential constraint ORDERS_CUSTOMER_ID_FK in
IBM20.

Insert statement will fail to run as each entry in the orders (customer_id) must have
an entry that is also a part of the customers (customer_id).
--------------------------------------------------------------------------------

/* Default constraints */
ALTER TABLE orders
       ALTER COLUMN order_date
       SET DEFAULT CURRENT_DATE;

Working statement
INSERT INTO orders VALUES (2001, DEFAULT, 111, 1001);

Result
ORDER_ID     ORDER_DATE   ASSOCIATE_ID CUSTOMER_ID
 -------     ----------   ------------ -----------
  2001       11/22/16         111         1001

Insert statement will insert the current date to the order_date attribute by default.
--------------------------------------------------------------------------------

/* Check constraints */
ALTER TABLE orders
       ADD CONSTRAINT orders_order_date_ck
       CHECK ( order_date >= '1980-01-01' );

Working statement
INSERT INTO orders
VALUES (2002, '2016-10-06', 222, 1003);

Failing statement
INSERT INTO orders
VALUES (2003, '1916-10-05', 222, 1004);

[SQL0545] INSERT, UPDATE, or MERGE not allowed by CHECK constraint.

Insert statement will fail to run as each order_date must be later than 1980-01-01.
--------------------------------------------------------------------------------
```

Products Table

```
DROP TABLE products;

CREATE TABLE products (
	product_id		INTEGER		NOT NULL,
	model			VARCHAR(10)	NOT NULL,
	brand_id		INTEGER		NOT NULL,
	category_id		INTEGER		NOT NULL,
	color_id		INTEGER		NOT NULL,
	width			INTEGER,
	depth			INTEGER,
	height			INTEGER,
	weight			INTEGER,
	powers			INTEGER,
	applicable_area		INTEGER,
	humidification_capacity	INTEGER,
	manufacture_date	DATE,
	item_price		DECIMAL(7,2)	NOT NULL
);

/* Primary key constraints */
ALTER TABLE products
	ADD CONSTRAINT products_product_id_pk
	PRIMARY KEY ( product_id );

Working statement
INSERT INTO products
VALUES (1001, 'L001', 101, 1, 22, 10, 11, 29, 2, 25, 115, 200,'2016-09-27', 239);

INSERT INTO products
VALUES (1002, 'L002', 101, 3, 11, 21, 22, 35, 4, 40, 250, 500,'2016-08-22', 459);

Failing statement
INSERT INTO products
VALUES (1001, 'L001', 101, 1, 22, 10, 11, 29, 2, 25, 115, 200,'2016-09-27', 239);

INSERT INTO products
VALUES (1002, 'L002', 101, 3, 11, 21, 22, 35, 4, 40, 250, 500,'2016-08-22', 459);

[SQL0803] Duplicate key value specified.

Insert statement will fail to run as a duplicate product_id was input.
--------------------------------------------------------------------------------

/* Unique key constraints */
ALTER TABLE products
	ADD CONSTRAINT products_model_uq
	UNIQUE ( model );

Working statement
INSERT INTO products
VALUES (1003, 'L003', 101, 1, 22, 10, 11, 29, 2, 25, 115, 200,'2016-09-27', 239);
```

```
Failing statement
INSERT INTO products
VALUES (1004, 'L001', 101, 1, 22, 10, 11, 29, 2, 25, 115, 200,'2016-09-27', 239);

[SQL0803] Duplicate key value specified.

Insert statement will fail to run as each product should have a unique model.
--------------------------------------------------------------------------------

/* Foreign key constraints */
ALTER TABLE products
       ADD CONSTRAINT products_brand_id_fk
       FOREIGN KEY ( brand_id )
       REFERENCES brands ( brand_id );

Working statement
INSERT INTO products
VALUES (1005, 'L005', 101, 1, 22, 10, 11, 29, 2, 25, 115, 200,'2016-09-27', 239);

Failing statement
INSERT INTO products
VALUES (1006, 'L006', 222, 1, 22, 10, 11, 29, 2, 25, 115, 200,'2016-09-27', 239);

[SQL0530] Operation not allowed by referential constraint PRODUCTS_BRAND_ID_FK in
IBM20.

Insert statement will fail to run as each entry in the products (brand_id) must have
an entry that is also a part of the brands (brand_id).
--------------------------------------------------------------------------------

ALTER TABLE products
       ADD CONSTRAINT products_category_id_fk
       FOREIGN KEY ( category_id )
       REFERENCES categories ( category_id );

Working statement
INSERT INTO products
VALUES (1007, 'L007', 101, 1, 22, 10, 11, 29, 2, 25, 115, 200,'2016-09-27', 239);

Failing statement
INSERT INTO products
VALUES (1008, 'L008', 101, 4, 22, 10, 11, 29, 2, 25, 115, 200,'2016-09-27', 239);

[SQL0530] Operation not allowed by referential constraint PRODUCTS_CATEGORY_ID_FK in
IBM20.

Insert statement will fail to run as each entry in the products (category_id) must
have an entry that is also a part of the categories (category_id).
--------------------------------------------------------------------------------

ALTER TABLE products
       ADD CONSTRAINT products_color_id_fk
       FOREIGN KEY ( color_id )
       REFERENCES colors ( color_id );
```

```
Working statement
INSERT INTO products
VALUES (1009, 'L009', 101, 1, 33, 10, 11, 29, 2, 25, 115, 200,'2016-09-27', 239);

Failing statement
INSERT INTO products
VALUES (1010, 'L010', 101, 1, 13, 10, 11, 29, 2, 25, 115, 200,'2016-09-27', 239);

[SQL0530] Operation not allowed by referential constraint PRODUCTS_COLOR_ID_FK in
IBM20.

Insert statement will fail to run as each entry in the products (color_id) must have
an entry that is also a part of the colors (color_id).
```

```
/* Default constraints */
ALTER TABLE products
       ALTER COLUMN manufacture_date
       SET DEFAULT CURRENT_DATE;

Working statement
INSERT INTO products
VALUES (1010, 'L010', 101, 1, 33, 10, 11, 29, 2, 25, 115, 200, DEFAULT, 239);

Result
PRODUCT_ID   MANUFACTURE_DATE    ITEM_PRICE
 -------     ---------------     ------------
  1010          11/22/16            111

Insert statement will insert the current date to the manufactrue_date attribute by
default.
```

```
/* Check constraints */
ALTER TABLE products
       ADD CONSTRAINT products_product_id_ck
       CHECK ( product_id >= 1000 AND product_id <= 9999 );

Working statement
INSERT INTO products
VALUES (1011, 'L011', 101, 1, 33, 10, 11, 29, 2, 25, 115, 200, DEFAULT, 239);

Failing statement
INSERT INTO products
VALUES (111, 'L012', 101, 1, 33, 10, 11, 29, 2, 25, 115, 200, DEFAULT, 239);

[SQL0545] INSERT, UPDATE, or MERGE not allowed by CHECK constraint.

Insert statement will fail to run as each product_id must be between 1000 and 9999.
```

```
ALTER TABLE products
       ADD CONSTRAINT products_width_ck
       CHECK ( width > 0 );
```

```
Working statement
INSERT INTO products
VALUES (1012, 'L012', 101, 1, 33, 10, 11, 29, 2, 25, 115, 200, DEFAULT, 239);

Failing statement
INSERT INTO products
VALUES (1013, 'L013', 101, 1, 33, 0, 11, 29, 2, 25, 115, 200, DEFAULT, 239);

[SQL0545] INSERT, UPDATE, or MERGE not allowed by CHECK constraint.

Insert statement will fail to run as each width must be greater than 0.
```

```
ALTER TABLE products
       ADD CONSTRAINT products_depth_ck
       CHECK ( depth > 0 );

Working statement
INSERT INTO products
VALUES (1013, 'L013', 101, 1, 33, 10, 11, 29, 2, 25, 115, 200, DEFAULT, 239);

Failing statement
INSERT INTO products
VALUES (1014, 'L014', 101, 1, 33, 10, 0, 29, 2, 25, 115, 200, DEFAULT, 239);

[SQL0545] INSERT, UPDATE, or MERGE not allowed by CHECK constraint.

Insert statement will fail to run as each depth must be greater than 0.
```

```
ALTER TABLE products
       ADD CONSTRAINT products_height_ck
       CHECK ( height > 0 );

Working statement
INSERT INTO products
VALUES (1014, 'L014', 101, 1, 33, 10, 11, 29, 2, 25, 115, 200, DEFAULT, 239);

Failing statement
INSERT INTO products
VALUES (1015, 'L015', 101, 1, 33, 10, 11, 0, 2, 25, 115, 200, DEFAULT, 239);

[SQL0545] INSERT, UPDATE, or MERGE not allowed by CHECK constraint.

Insert statement will fail to run as each height must be greater than 0.
```

```
ALTER TABLE products
       ADD CONSTRAINT products_weight_ck
       CHECK ( weight > 0 );

Working statement
INSERT INTO products
VALUES (1015, 'L015', 101, 1, 33, 10, 11, 29, 2, 25, 115, 200, DEFAULT, 239);
```

```
Failing statement
INSERT INTO products
VALUES (1016, 'L016', 101, 1, 33, 10, 11, 29, 0, 25, 115, 200, DEFAULT, 239);

[SQL0545] INSERT, UPDATE, or MERGE not allowed by CHECK constraint.

Insert statement will fail to run as each weight must be greater than 0.
--------------------------------------------------------------------------------

ALTER TABLE products
      ADD CONSTRAINT products_powers_ck
      CHECK ( powers >= 15 AND powers <= 40 );

Working statement
INSERT INTO products
VALUES (1016, 'L016', 101, 1, 33, 10, 11, 29, 2, 25, 115, 200, DEFAULT, 239);

Failing statement
INSERT INTO products
VALUES (1017, 'L017', 101, 1, 33, 10, 11, 29, 2, 45, 115, 200, DEFAULT, 239);

[SQL0545] INSERT, UPDATE, or MERGE not allowed by CHECK constraint.

Insert statement will fail to run as each powers must be between 15 and 40.
--------------------------------------------------------------------------------

ALTER TABLE products
      ADD CONSTRAINT products_applicable_area_ck
      CHECK ( applicable_area > 0 );

Working statement
INSERT INTO products
VALUES (1017, 'L017', 101, 1, 33, 10, 11, 29, 2, 25, 115, 200, DEFAULT, 239);

Failing statement
INSERT INTO products
VALUES (1018, 'L018', 101, 1, 33, 10, 11, 29, 2, 25, 0, 200, DEFAULT, 239);

[SQL0545] INSERT, UPDATE, or MERGE not allowed by CHECK constraint.

Insert statement will fail to run as each applicable_area must be greater than 0.
--------------------------------------------------------------------------------

ALTER TABLE products
      ADD CONSTRAINT products_humidification_capacity_ck
      CHECK ( humidification_capacity > 0 );

Working statement
INSERT INTO products
VALUES (1018, 'L018', 101, 1, 33, 10, 11, 29, 2, 25, 115, 200, DEFAULT, 239);

Failing statement
INSERT INTO products
VALUES (1019, 'L019', 101, 1, 33, 10, 11, 29, 2, 25, 115, 0, DEFAULT, 239);
```

```
[SQL0545] INSERT, UPDATE, or MERGE not allowed by CHECK constraint.

Insert statement will fail to run as each humidification_capacity must be greater
than 0.
--------------------------------------------------------------------------------
```

```
ALTER TABLE products
      ADD CONSTRAINT products_item_price_ck
      CHECK ( item_price > 0 );

Working statement
INSERT INTO products
VALUES (1019, 'L019', 101, 1, 33, 10, 11, 29, 2, 25, 115, 200, DEFAULT, 239);

Failing statement
INSERT INTO products
VALUES (1020, 'L020', 101, 1, 33, 10, 11, 29, 2, 25, 115, 200, DEFAULT, 0);

[SQL0545] INSERT, UPDATE, or MERGE not allowed by CHECK constraint.

Insert statement will fail to run as each item_price must be greater than 0.
--------------------------------------------------------------------------------
```

Order_products Table

```
DROP TABLE order_products;

CREATE TABLE order_products (
      order_id            INTEGER             NOT NULL,
      product_id          INTEGER             NOT NULL,
      quantity            INTEGER             NOT NULL
);

/* Primary key constraints */
ALTER TABLE order_products
      ADD CONSTRAINT order_products_pk
      PRIMARY KEY ( order_id, product_id );

Working statement
INSERT INTO order_products
VALUES (1001, 1002, 1);

Failing statement
INSERT INTO order_products
VALUES (1001, 1002, 1);

[SQL0803] Duplicate key value specified.

Insert statement will fail to run as a duplicate order_id and product_id were input.
--------------------------------------------------------------------------------
```

```
/* Foreign key constraints */
ALTER TABLE order_products
       ADD CONSTRAINT order_products_order_id_fk
       FOREIGN KEY (order_id)
       REFERENCES orders(order_id);

Working statement
INSERT INTO order_products
VALUES (1002, 1002, 1);

Failing statement
INSERT INTO order_products
VALUES (2222, 1002, 1);

[SQL0530] Operation not allowed by referential constraint ORDER_PRODUCTS_ORDER_ID_FK
in IBM20.

Insert statement will fail to run as each entry in the order_products (order_id) must
have an entry that is also a part of the orders (order_id).
--------------------------------------------------------------------------------

ALTER TABLE order_products
       ADD CONSTRAINT order_products_product_id_fk
       FOREIGN KEY ( product_id )
       REFERENCES products ( product_id );

Working statement
INSERT INTO order_products
VALUES (1003, 1002, 1);

Failing statement
INSERT INTO order_products
VALUES (1003, 2222, 1);

[SQL0530] Operation not allowed by referential constraint
ORDER_PRODUCTS_PRODUCT_ID_FK in IBM20.

Insert statement will fail to run as each entry in the order_products (product_id)
must have an entry that is also a part of the products (product_id).
--------------------------------------------------------------------------------

/* Check constraints */
ALTER TABLE order_products
       ADD CONSTRAINT invoices_quantity_ck
       CHECK ( quantity > 0 );

Working statement
INSERT INTO order_products
VALUES (1004, 1002, 1);

Failing statement
INSERT INTO order_products
VALUES (1005, 1002, 0);

[SQL0545] INSERT, UPDATE, or MERGE not allowed by CHECK constraint.
```

```
Insert statement will fail to run as each quantity must be greater than 0.
--------------------------------------------------------------------------------
```

Colors Table

```
DROP TABLE colors;

CREATE TABLE colors (
	color_id		INTEGER			NOT NULL,
	color_name		VARCHAR(20)		NOT NULL
);

/* Primary key constraints */
ALTER TABLE colors
	ADD CONSTRAINT colors_pk
	PRIMARY KEY ( color_id );

Working statement
INSERT INTO colors
VALUES (11, 'white');

Failing statement
INSERT INTO colors
VALUES (11, 'white');

[SQL0803] Duplicate key value specified.

Insert statement will fail to run as a duplicate color_id was input.
--------------------------------------------------------------------------------

/* Unique key constraints */
ALTER TABLE colors
	ADD CONSTRAINT colors_color_name_uq
	UNIQUE ( color_name );

Working statement
INSERT INTO colors
VALUES (22, 'black');

Failing statement
INSERT INTO colors
VALUES (22, 'white');

[SQL0803] Duplicate key value specified.

Insert statement will fail to run as each color must have a unique color name.
--------------------------------------------------------------------------------
```

Categories Table

```
DROP TABLE categories;
```

```
CREATE TABLE categories (
	category_id		INTEGER			NOT NULL,
	category_name		VARCHAR(20)		NOT NULL
);

/* Primary key constraints */
ALTER TABLE categories
	ADD CONSTRAINT categories_pk
	PRIMARY KEY ( category_id );

Working statement
INSERT INTO categories
VALUES (1, 'household');

Failing statement
INSERT INTO categories
VALUES (1, 'household');

[SQL0803] Duplicate key value specified.

Insert statement will fail to run as a duplicate category_id was input.
--------------------------------------------------------------------------------

/* Unique key constraints */
ALTER TABLE categories
	ADD CONSTRAINT categories_category_name_uq
	UNIQUE ( category_name );

Working statement
INSERT INTO categories
VALUES (2, 'commercial');

Failing statement
INSERT INTO categories
VALUES (3, 'household');

[SQL0803] Duplicate key value specified.

Insert statement will fail to run as each category must have a unique category name.
--------------------------------------------------------------------------------

/* Check constraints */
ALTER TABLE categories
	ADD CONSTRAINT categories_category_name_ck
	CHECK (category_name IN ('household', 'commercial', 'industrial'));

Working statement
INSERT INTO categories
VALUES (3, 'industrial');

Failing statement
INSERT INTO categories
VALUES (4, 'industry');
```

```
[SQL0545] INSERT, UPDATE, or MERGE not allowed by CHECK constraint.
```

Insert statement will fail to run as each category must be in “household”, “commercial” and “industrial”.

Brands Table

```
DROP TABLE brands;

CREATE TABLE brands (
	brand_id		INTEGER		NOT NULL,
	brand_name		VARCHAR(20)	NOT NULL
);

/* Primary key constraints */
ALTER TABLE brands
	ADD CONSTRAINT brands_pk
	PRIMARY KEY ( brand_id );

Working statement
INSERT INTO brands
VALUES (101, 'LG');

Failing statement
INSERT INTO brands
VALUES (101, 'LG');

[SQL0803] Duplicate key value specified.
```

Insert statement will fail to run as a duplicate brand_id was input.

```
/* Unique key constraints */
ALTER TABLE brands
	ADD CONSTRAINT brands_brand_name_uq
	UNIQUE ( brand_name );

Working statement
INSERT INTO brands
VALUES (102, 'SONY');

Failing statement
INSERT INTO brands
VALUES (103, 'LG');

[SQL0803] Duplicate key value specified.
```

Insert statement will fail to run as each brand must have a unique brand name.
