# ITEC 4200 Advanced Database Semester Project

Student Name: **Robert Bryan**

Semester: **Fall 2016**

Name of Project: **Robert’s Custom Computer Build Company Database**

Email: **rbryan3@ggc.edu**

Phone number(s): **(770) - 733 - 5936**

Part 1: Executive Summary

This database contains information about Robert’s Custom Computer Build Company, including: customer information, computer builds (and the parts that compose them), the builds in which a customer has purchased, and information about pertaining to the available computer parts for the builds (RAM, CPU, GPU, etc.). Whether a customer selects from a previously completed build, or orders a new build, the database’s design will account for both. The database will be implemented using Oracle 11g Express Database.

The implementation of the database contains eleven tables that provide information about customers, the builds in which a customer has purchased, the computer builds, and the parts that a computer build can contain. The database tables include the following: ***Customer***, ***Builds\_Purchased***, ***Computer\_Build***, ***CPU***, ***Hard\_Drive***, ***Tower***, ***GPU***, ***Motherboard***, ***Power\_Supply***, ***Solid\_State\_Drive***, and ***RAM***. These tables represent the main entities of the application, and contain links that define the relationships between entities.

The information in the ***Customer*** table includes the customer’s first and last name, street address, zip code, state, and phone. The ***Builds\_Purchased*** table includes the one or more builds a customer has purchased and the date in which the transaction took place. The ***Computer\_Build*** table includes the parts that make up the build. The ***CPU*** includes information about the different Central Processing Units that a build can contain, including: its speed, processor count, brand, series, socket type, and cost price. The ***Hard\_Drive*** table includes information about the different hard drives that a build can contain, including: its storage capacity, brand, series, speed, and cost price. The ***Tower*** table includes information about the different towers that a build can contain, including: its size, color, brand, series, and cost price. The ***GPU*** table includes information about the different Graphical Processing Units that a build can contain, including: its screen resolution, graphical ram size, base and boost clock, brand, series, and cost price.

The ***Motherboard*** table includes information about the different motherboards that a build can contain, including: its chipset type, RAM memory technology, number of memory slots, brand, series, and cost price. The ***Power\_Supply*** table includes information about the different power supplies that a build can contain, including: its brand, series, wattage, and cost price. The ***Solid\_State\_Drive*** table includes information about the different solid state drives that a build can contain, including: its storage capacity, brand, series, and cost price. The ***RAM*** table includes information about the different RAM stick packages that a build can contain, including: its storage capacity, memory type, color, number of sticks, brand, series, and cost price. The customer can purchase multiple computer builds, and a computer build can be purchased by multiple customers. The computer build must contain one and only one CPU, tower, motherboard, power supply, and RAM (one set of RAM sticks). The computer build must contain either one hard drive or one SSD, but it may contain both. The computer may or may not include a GPU.

This database organizes information about Robert’s Custom Computer Building Company in a way that allows for queries to easily to retrieve a great deal of information, such as the total profit the company has made, or the best-selling builds. Included are several queries that show the database’s functionality, and a report that displays the data in a more presentable way.

Part 2: Database Schema

**CUSTOMER**(Customer\_ID, First\_name, Last\_name, Street\_address, Zip\_code, State, Phone)

**BUILDS\_PURCHASED**(Build\_ID, Customer\_ID, Purchase\_date)

FK Build\_ID 🡪 COMPUTER\_BUILD

FK Customer\_ID 🡪 CUSTOMER

**COMPUTER\_BUILD**(Build\_ID, Tower\_ID, MB\_ID, PS\_ID, RAM\_ID, CPU\_ID, GPU\_ID, SSD\_ID, HDD\_ID)

FK Tower\_ID 🡪 TOWER

FK MB\_ID 🡪 MOTHERBOARD

FK PS\_ID 🡪 POWER\_SUPPLY

FK RAM\_ID 🡪 RAM

FK CPU\_ID 🡪 CPU

FK GPU\_ID 🡪 GPU

FK SSD\_ID 🡪 SSD

FK HDD\_ID 🡪 HARD\_DRIVE

**CPU**(CPU\_ID, CPU\_speed, CPU\_count, CPU\_brand, CPU\_series, CPU\_socket, Cost\_price)

**HARD\_DRIVE**(HDD\_ID, HDD\_storage\_capacity, HDD\_brand, HDD\_series, HDD\_speed, Cost\_price)

**TOWER**(Tower\_ID, Tower\_size, Tower\_color, Tower\_brand, Tower\_series, Cost\_price)

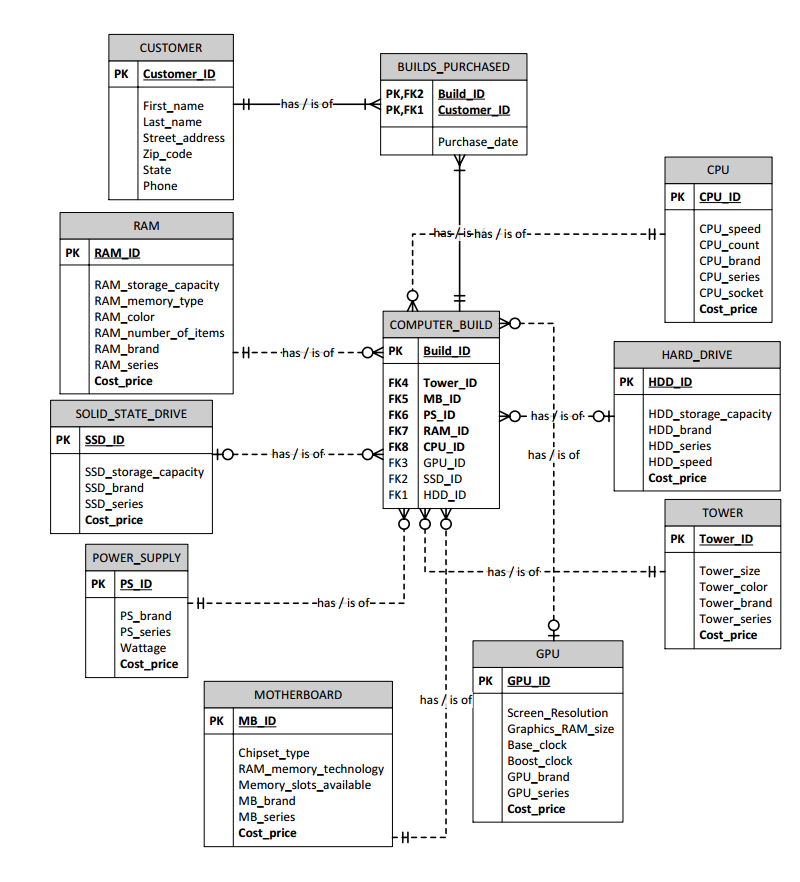
**GPU**(GPU\_ID, Screen\_resolution, Graphics\_RAM\_size, Base\_clock, Boost\_clock, GPU\_brand, GPU\_series, Cost\_price)

**MOTHERBOARD**(MB\_ID, Chipset\_type, RAM\_memory\_technology, Memory\_slots\_available, MB\_brand, MB\_series, Cost\_price)

**POWER\_SUPPLY**(PS\_ID, PS\_brand, PS\_series, Wattage, Cost\_price)

**SOLID\_STATE\_DRIVE**(SSD\_ID, SSD\_storage\_capacity, SSD\_brand, SSD\_series, Cost\_price)

**RAM**(RAM\_ID, RAM\_storage\_capacity, RAM\_memory\_type, RAM\_color, RAM\_number\_of\_items, RAM\_brand, RAM\_series, Cost\_price)

Part 2: ER Diagram

Part 3: Oracle Implementation

-- Initial Drops

DROP TABLE customer CASCADE CONSTRAINTS;

DROP TABLE ram CASCADE CONSTRAINTS;

DROP TABLE solid\_state\_drive CASCADE CONSTRAINTS;

DROP TABLE power\_supply CASCADE CONSTRAINTS;

DROP TABLE motherboard CASCADE CONSTRAINTS;

DROP TABLE gpu CASCADE CONSTRAINTS;

DROP TABLE tower CASCADE CONSTRAINTS;

DROP TABLE hard\_drive CASCADE CONSTRAINTS;

DROP TABLE cpu CASCADE CONSTRAINTS;

DROP TABLE computer\_build CASCADE CONSTRAINTS;

DROP TABLE builds\_purchased CASCADE CONSTRAINTS;

DROP SEQUENCE computer\_build\_build\_id\_seq;

DROP SEQUENCE cpu\_cpu\_id\_seq;

DROP SEQUENCE hard\_drive\_hdd\_id\_seq;

DROP SEQUENCE tower\_tower\_id\_seq;

DROP SEQUENCE gpu\_gpu\_id\_seq;

DROP SEQUENCE motherboard\_mb\_id\_seq;

DROP SEQUENCE power\_supply\_ps\_id\_seq;

DROP SEQUENCE solid\_state\_drive\_ssd\_id\_seq;

DROP SEQUENCE ram\_ram\_id\_seq;

DROP SEQUENCE customer\_customer\_id\_seq;

DROP SEQUENCE bp\_build\_purchased\_id\_seq;

--Table create statements

CREATE TABLE CUSTOMER

( customer\_id NUMBER,

first\_name VARCHAR2(40),

last\_name VARCHAR2(40),

street\_address VARCHAR2(60),

state CHAR(2) DEFAULT 'GA',

phone\_number VARCHAR2(20),

CONSTRAINT customer\_customer\_id\_pk PRIMARY KEY(customer\_id)

);

CREATE TABLE ram

( ram\_id NUMBER

, ram\_storage\_capacity VARCHAR2(10)

, ram\_memory\_type VARCHAR2(20)

, ram\_memory\_technology VARCHAR2(20)

, ram\_color VARCHAR2(20)

, ram\_number\_of\_items NUMBER

, ram\_brand VARCHAR2(20)

, ram\_series VARCHAR2(20)

, cost\_price NUMBER(6, 2)

, CONSTRAINT ram\_ram\_id\_pk PRIMARY KEY (ram\_id)

);

CREATE TABLE solid\_state\_drive

( ssd\_id NUMBER

, ssd\_storage\_capacity VARCHAR2(20)

, ssd\_brand VARCHAR2(20)

, ssd\_series VARCHAR2(20)

, cost\_price NUMBER(6, 2)

, CONSTRAINT solid\_state\_drive\_ssd\_id\_pk PRIMARY KEY (ssd\_id)

);

CREATE TABLE power\_supply

( ps\_id NUMBER

, ps\_brand VARCHAR2(20)

, ps\_series VARCHAR2(20)

, wattage NUMBER

, cost\_price NUMBER

, CONSTRAINT power\_supply\_ps\_id\_pk PRIMARY KEY (ps\_id)

);

CREATE TABLE motherboard

( mb\_id NUMBER

, chipset\_type VARCHAR2(20)

, ram\_memory\_technology VARCHAR2(20)

, memory\_slots\_available NUMBER

, mb\_brand VARCHAR2(20)

, mb\_series VARCHAR2(30)

, cost\_price NUMBER

, CONSTRAINT motherboard\_mb\_id\_pk PRIMARY KEY (mb\_id)

);

CREATE TABLE gpu

( gpu\_id NUMBER

, screen\_resolution VARCHAR2(20)

, graphics\_ram\_size VARCHAR2(20)

, base\_clock VARCHAR2(20)

, boost\_clock VARCHAR2(20)

, gpu\_brand VARCHAR(30)

, gpu\_series VARCHAR2(30)

, cost\_price NUMBER

, CONSTRAINT gpu\_gpu\_id\_pk PRIMARY KEY (gpu\_id)

);

CREATE TABLE tower

( tower\_id NUMBER

, tower\_size VARCHAR2(20)

, tower\_color VARCHAR2(20)

, tower\_brand VARCHAR2(20)

, tower\_series VARCHAR2(20)

, cost\_price NUMBER

, CONSTRAINT tower\_tower\_id\_pk PRIMARY KEY (tower\_id)

);

CREATE TABLE hard\_drive

( hdd\_id NUMBER

, hdd\_storage\_capacity VARCHAR2(20)

, hdd\_brand VARCHAR2(20)

, hdd\_series VARCHAR2(20)

, cost\_price NUMBER

, CONSTRAINT hard\_drive\_hdd\_id\_pk PRIMARY KEY (hdd\_id)

);

CREATE TABLE cpu

( cpu\_id NUMBER

, cpu\_speed VARCHAR2(20)

, cpu\_count VARCHAR2(20)

, cpu\_brand VARCHAR2(20)

, cpu\_series VARCHAR2(20)

, cpu\_socket VARCHAR2(20)

, cost\_price NUMBER

, CONSTRAINT cpu\_cpu\_id\_pk PRIMARY KEY (cpu\_id)

);

CREATE TABLE computer\_build

( build\_id NUMBER

, tower\_id NUMBER

CONSTRAINT computer\_build\_tower\_id\_nn NOT NULL

, mb\_id NUMBER

CONSTRAINT computer\_build\_mb\_id\_nn NOT NULL

, ps\_id NUMBER

CONSTRAINT computer\_build\_ps\_id\_nn NOT NULL

, ram\_id NUMBER

CONSTRAINT computer\_build\_ram\_id\_nn NOT NULL

, cpu\_id NUMBER

CONSTRAINT computer\_build\_cpu\_id\_nn NOT NULL

, gpu\_id NUMBER

, ssd\_id NUMBER

, hdd\_id NUMBER

, CONSTRAINT computer\_build\_build\_id\_pk PRIMARY KEY (build\_id)

, CONSTRAINT tower\_id\_fk FOREIGN KEY (tower\_id) REFERENCES tower

, CONSTRAINT mb\_id\_fk FOREIGN KEY (mb\_id) REFERENCES motherboard

, CONSTRAINT ps\_id\_fk FOREIGN KEY (ps\_id) REFERENCES power\_supply

, CONSTRAINT ram\_id\_fk FOREIGN KEY (ram\_id) REFERENCES ram

, CONSTRAINT cpu\_id\_fk FOREIGN KEY (cpu\_id) REFERENCES cpu

, CONSTRAINT gpu\_id\_fk FOREIGN KEY (gpu\_id) REFERENCES gpu

, CONSTRAINT ssd\_id\_fk FOREIGN KEY (ssd\_id) REFERENCES solid\_state\_drive

, CONSTRAINT hdd\_id\_fk FOREIGN KEY (hdd\_id) REFERENCES hard\_drive

);

CREATE TABLE builds\_purchased

( build\_purchased\_id NUMBER

, build\_id NUMBER

CONSTRAINT bp\_build\_id\_nn NOT NULL

, customer\_id NUMBER

CONSTRAINT bp\_customer\_id\_nn NOT NULL

, purchase\_date DATE

CONSTRAINT bp\_purchase\_date\_nn NOT NULL

, CONSTRAINT bp\_purchased\_id\_pk PRIMARY KEY (build\_purchased\_id)

, CONSTRAINT build\_id\_fk FOREIGN KEY (build\_id) REFERENCES computer\_build

, CONSTRAINT customer\_id\_fk FOREIGN KEY (customer\_id) REFERENCES customer

);

--Sequence create statements

CREATE SEQUENCE customer\_customer\_id\_seq;

CREATE SEQUENCE ram\_ram\_id\_seq;

CREATE SEQUENCE solid\_state\_drive\_ssd\_id\_seq;

CREATE SEQUENCE power\_supply\_ps\_id\_seq;

CREATE SEQUENCE motherboard\_mb\_id\_seq;

CREATE SEQUENCE gpu\_gpu\_id\_seq;

CREATE SEQUENCE tower\_tower\_id\_seq;

CREATE SEQUENCE cpu\_cpu\_id\_seq;

CREATE SEQUENCE hard\_drive\_hdd\_id\_seq;

CREATE SEQUENCE computer\_build\_build\_id\_seq;

CREATE SEQUENCE bp\_build\_purchased\_id\_seq;

--Customer table inserts

INSERT INTO customer VALUES

(customer\_customer\_id\_seq.nextval, 'Bob', 'Green', '4312 Heartsfield Lane', 'GA', '432-432-5423');

INSERT INTO customer VALUES

(customer\_customer\_id\_seq.nextval, 'Kate', 'Smith', '2313 Tornado Drive', 'CA', '321-412-1231');

INSERT INTO customer VALUES

(customer\_customer\_id\_seq.nextval, 'Zac', 'Toe', '3212 Safe Round', 'CA', '332-333-5423');

INSERT INTO customer VALUES

(customer\_customer\_id\_seq.nextval, 'Faith', 'Smoltz', '2314 Life Savers Corner', 'GA', '102-442-1213');

INSERT INTO customer VALUES

(customer\_customer\_id\_seq.nextval, 'Sarah', 'Twinkle', '1246 Green Smith Lane', 'GA', '770-432-4322');

INSERT INTO customer VALUES

(customer\_customer\_id\_seq.nextval, 'Freddie', 'Mars', '8751 Dunkin'' Donuts Corner', 'GA',

'232-123-4321');

--Ram table inserts

INSERT INTO ram VALUES

(ram\_ram\_id\_seq.nextval, '8 GB', 'DDR3 SDRAM', 'DDR3 1600', 'Black', 1, 'Corsair',

'Vengeance', 45);

INSERT INTO ram VALUES

(ram\_ram\_id\_seq.nextval, '16 GB', 'DDR3 SDRAM', 'DDR3 1800', 'Black', 2, 'Kingston',

'HyperX FURY', 80);

INSERT INTO ram VALUES

(ram\_ram\_id\_seq.nextval, '8 GB', 'DDR3 SDRAM', 'DDR3 1800', 'Red', 2, 'Kingston',

'HyperX FURY', 55);

INSERT INTO ram VALUES

(ram\_ram\_id\_seq.nextval, '32 GB', 'DDR4 SDRAM', 'DDR4 2400', 'Red', 2, 'Ballistix',

'Sport LT', 170);

INSERT INTO ram VALUES

(ram\_ram\_id\_seq.nextval, '64 GB', 'DDR4 SDRAM', 'DDR4 2400', 'White', 4, 'Ballistix',

'Sport LT', 300);

--SSD table inserts

INSERT INTO solid\_state\_drive VALUES

(solid\_state\_drive\_ssd\_id\_seq.nextval, '500 GB', 'Samsung', 'EVO', 157.10);

INSERT INTO solid\_state\_drive VALUES

(solid\_state\_drive\_ssd\_id\_seq.nextval, '1000 GB', 'Samsung', 'EVO', 499.99);

INSERT INTO solid\_state\_drive VALUES

(solid\_state\_drive\_ssd\_id\_seq.nextval, '250 GB', 'Samsung', 'EVO', 95.00);

INSERT INTO solid\_state\_drive VALUES

(solid\_state\_drive\_ssd\_id\_seq.nextval, '120 GB', 'Kingston', 'Digital', 44.95);

INSERT INTO solid\_state\_drive VALUES

(solid\_state\_drive\_ssd\_id\_seq.nextval, '240 GB', 'Kingston', 'Digital', 73.99);

INSERT INTO solid\_state\_drive VALUES

(solid\_state\_drive\_ssd\_id\_seq.nextval, '480 GB', 'Kingston', 'Digital', 139.99);

--Power supply table inserts

INSERT INTO power\_supply VALUES

(power\_supply\_ps\_id\_seq.nextval, 'EVGA', 'W1', 500, 35.99);

INSERT INTO power\_supply VALUES

(power\_supply\_ps\_id\_seq.nextval, 'EVGA', 'B1', 700, 53.82);

INSERT INTO power\_supply VALUES

(power\_supply\_ps\_id\_seq.nextval, 'EVGA', 'G2', 750, 105.90);

INSERT INTO power\_supply VALUES

(power\_supply\_ps\_id\_seq.nextval, 'EVGA', 'G2', 1600, 308.26);

INSERT INTO power\_supply VALUES

(power\_supply\_ps\_id\_seq.nextval, 'EVGA', 'G2', 850, 119.32);

INSERT INTO power\_supply VALUES

(power\_supply\_ps\_id\_seq.nextval, 'Corsair', 'CX', 500, 49.99);

INSERT INTO power\_supply VALUES

(power\_supply\_ps\_id\_seq.nextval, 'Corsair', 'CX', 600, 54.99);

INSERT INTO power\_supply VALUES

(power\_supply\_ps\_id\_seq.nextval, 'Corsair', 'CX', 750, 59.99);

--Motherboard table inserts

INSERT INTO motherboard VALUES

(motherboard\_mb\_id\_seq.nextval, 'AMD 970', 'DDR3 2133', 4, 'MSI', '970 GAMING', 84.99);

INSERT INTO motherboard VALUES

(motherboard\_mb\_id\_seq.nextval, 'AMD 970', 'DDR3 1066', 4, 'Gigabyte', 'AM3+', 149.99);

INSERT INTO motherboard VALUES

(motherboard\_mb\_id\_seq.nextval, 'Intel Z97', 'DDR3 1066', 4, 'MSI', 'Intel Z97 LGA', 89.00);

INSERT INTO motherboard VALUES

(motherboard\_mb\_id\_seq.nextval, 'Intel Z170', 'DDR4 2667', 4, 'ASUS', 'TUF', 179.99);

INSERT INTO motherboard VALUES

(motherboard\_mb\_id\_seq.nextval, 'Intel Z170', 'DDR4 3000', 4, 'ASUS', 'ROG MAXIMUS VIII', 253.99);

--GPU table inserts

INSERT INTO gpu VALUES

(gpu\_gpu\_id\_seq.nextval, '4096 x 2160', '4 GB', '1165 MHz', '1317 MHz', 'EVGA',

'GeForce GTX 970', 399.99);

INSERT INTO gpu VALUES

(gpu\_gpu\_id\_seq.nextval, '4096 x 2160', '8 GB', null, '1835 MHz', 'ASUS',

'GeForce GTX 1080', 669.99);

INSERT INTO gpu VALUES

(gpu\_gpu\_id\_seq.nextval, '2560 x 1600', '2 GB', '1176 MHz', '1255 MHz', 'EVGA',

'GeForce GTX 750Ti', 119.99);

INSERT INTO gpu VALUES

(gpu\_gpu\_id\_seq.nextval, '2560 x 1600', '4 GB', null, null, 'MSI', 'Gaming RX 480', 219.99);

INSERT INTO gpu VALUES

(gpu\_gpu\_id\_seq.nextval, '4096 x 2160', '6 GB', '1582 MHz', '1797 MHz', 'Gigabyte',

'GeForce GTX 106 WindForce', 119.99);

--Tower table inserts

INSERT INTO tower VALUES

(tower\_tower\_id\_seq.nextval, 'Full', 'Black', 'Phanteks', 'Enthoo Pro', 100.0);

INSERT INTO tower VALUES

(tower\_tower\_id\_seq.nextval, 'Full', 'White', 'Phanteks', 'Enthoo Pro', 110.0);

INSERT INTO tower VALUES

(tower\_tower\_id\_seq.nextval, 'Full', 'Titanium Green', 'Phanteks', 'Enthoo Pro', 110.0);

INSERT INTO tower VALUES

(tower\_tower\_id\_seq.nextval, 'Full', 'Black', 'NZXT', 'Source 530', 84.99);

INSERT INTO tower VALUES

(tower\_tower\_id\_seq.nextval, 'Full', 'Black', 'Corsair', 'Obsidian Series 750D', 155.49);

INSERT INTO tower VALUES

(tower\_tower\_id\_seq.nextval, 'Mid', 'Black', 'Corsair', 'Carbide Series 100R', 52.95);

INSERT INTO tower VALUES

(tower\_tower\_id\_seq.nextval, 'Mid', 'White', 'NZXT', 'Phantom 240', 64.99);

INSERT INTO tower VALUES

(tower\_tower\_id\_seq.nextval, 'Mini', 'Black', 'Rosewill', 'Micro ATX Mini', 39.99);

--Hard drive table inserts

INSERT INTO hard\_drive VALUES

(hard\_drive\_hdd\_id\_seq.nextval, '1000 GB', 'Western Digital', 'Blue', 49.99);

INSERT INTO hard\_drive VALUES

(hard\_drive\_hdd\_id\_seq.nextval, '4000 GB', 'Western Digital', 'Red', 149.99);

INSERT INTO hard\_drive VALUES

(hard\_drive\_hdd\_id\_seq.nextval, '2000 GB', 'Western Digital', 'Black', 124.99);

INSERT INTO hard\_drive VALUES

(hard\_drive\_hdd\_id\_seq.nextval, '3000 GB', 'Western Digital', 'Black', 168.59);

INSERT INTO hard\_drive VALUES

(hard\_drive\_hdd\_id\_seq.nextval, '6000 GB', 'Western Digital', 'Black', 279.99);

INSERT INTO hard\_drive VALUES

(hard\_drive\_hdd\_id\_seq.nextval, '500 GB', 'Western Digital', 'Blue', 41.80);

--CPU table inserts

INSERT INTO cpu VALUES

(cpu\_cpu\_id\_seq.nextval, '4.00 Ghz', 4, 'Intel', 'Core i7 6700k', 'LGA 1151', 329);

INSERT INTO cpu VALUES

(cpu\_cpu\_id\_seq.nextval, '4.00 Ghz', 6, 'Intel', 'Core i7 6800k', 'LGA 2011-v3', 439.5);

INSERT INTO cpu VALUES

(cpu\_cpu\_id\_seq.nextval, '4.1 Ghz', 4, 'Intel', 'Core i7-4790K', 'LGA 1150', 335.62);

INSERT INTO cpu VALUES

(cpu\_cpu\_id\_seq.nextval, '3.50 Ghz', 4, 'Intel', 'Core i5-4690K', 'LGA 1150', 235.99);

INSERT INTO cpu VALUES

(cpu\_cpu\_id\_seq.nextval, '2.80 Ghz', 4, 'AMD', 'A10 7870K', 'AM3', 144.99);

INSERT INTO cpu VALUES

(cpu\_cpu\_id\_seq.nextval, '4.0 Ghz', 8, 'AMD', 'FX-8350', 'AM3+', 149.99);

--Computer build table inserts

INSERT INTO computer\_build VALUES

(computer\_build\_build\_id\_seq.nextval, 1, 3, 2, 1, 1, 1, 2, null);

INSERT INTO computer\_build VALUES

(computer\_build\_build\_id\_seq.nextval, 5, 1, 4, 2, 6, 4, null, 5);

INSERT INTO computer\_build VALUES

(computer\_build\_build\_id\_seq.nextval, 3, 5, 8, 3, 3, 2, 1, 3);

INSERT INTO computer\_build VALUES

(computer\_build\_build\_id\_seq.nextval, 7, 4, 1, 4, 4, 3, 3, 1);

INSERT INTO computer\_build VALUES

(computer\_build\_build\_id\_seq.nextval, 4, 1, 7, 4, 5, 4, 2, 2);

INSERT INTO computer\_build VALUES

(computer\_build\_build\_id\_seq.nextval, 2, 3, 2, 5, 1, 1, 3, 2);

--Builds purchased table inserts

INSERT INTO builds\_purchased VALUES

(bp\_build\_purchased\_id\_seq.nextval, 1, 1, '02-MAR-2016');

INSERT INTO builds\_purchased VALUES

(bp\_build\_purchased\_id\_seq.nextval, 1, 2, '04-JUL-2015');

INSERT INTO builds\_purchased VALUES

(bp\_build\_purchased\_id\_seq.nextval, 4, 5, '11-MAR-2016');

INSERT INTO builds\_purchased VALUES

(bp\_build\_purchased\_id\_seq.nextval, 3, 4, '04-MAR-2016');

INSERT INTO builds\_purchased VALUES

(bp\_build\_purchased\_id\_seq.nextval, 3, 3, '04-MAR-2016');

INSERT INTO builds\_purchased VALUES

(bp\_build\_purchased\_id\_seq.nextval, 4, 2, '24-JAN-2016');

INSERT INTO builds\_purchased VALUES

(bp\_build\_purchased\_id\_seq.nextval, 3, 4, '19-OCT-2016');

INSERT INTO builds\_purchased VALUES

(bp\_build\_purchased\_id\_seq.nextval, 3, 3, '24-JUL-2016');

COMMIT;

Part 4: SQL Queries

-- Query 1

-- This query will display the first name, last name, address, and phone number of the

-- customers who reside in georgia and have purchased a build.

SELECT first\_name || ' ' || last\_name AS "Customer", street\_address AS "Address", phone\_number AS "Phone"

FROM customer

WHERE state LIKE 'GA' AND customer\_id IN (SELECT customer\_id FROM builds\_purchased)

ORDER BY 1;

-- Query 2

-- This query displays the total build cost for each of the available builds, and

-- assigns them a build type based on the cost.

SELECT TO\_CHAR(c.cost\_price + hd.cost\_price + t.cost\_price + g.cost\_price + m.cost\_price + ps.cost\_price + ssd.cost\_price + r.cost\_price, '$9999.99') AS "Total Build Cost (US $)",

CASE

WHEN c.cost\_price + hd.cost\_price + t.cost\_price + g.cost\_price + m.cost\_price + ps.cost\_price + ssd.cost\_price + r.cost\_price BETWEEN 0 AND 1000 THEN 'Starter Build'

WHEN c.cost\_price + hd.cost\_price + t.cost\_price + g.cost\_price + m.cost\_price + ps.cost\_price + ssd.cost\_price + r.cost\_price BETWEEN 1000 AND 1500 THEN 'High-end Build'

WHEN c.cost\_price + hd.cost\_price + t.cost\_price + g.cost\_price + m.cost\_price + ps.cost\_price + ssd.cost\_price + r.cost\_price BETWEEN 1500 AND 2000 THEN 'Super-end Build'

ELSE 'Extremely Expensive Build'

END AS "Build Types"

FROM computer\_build cb JOIN CPU c ON (cb.cpu\_id = c.cpu\_id)

JOIN hard\_drive hd ON (cb.hdd\_id = hd.hdd\_id)

JOIN tower t ON (cb.tower\_id = t.tower\_id)

JOIN gpu g ON (cb.gpu\_id = g.gpu\_id)

JOIN motherboard m ON (cb.mb\_id = m.mb\_id)

JOIN power\_supply ps ON (cb.ps\_id = ps.ps\_id)

JOIN solid\_state\_drive ssd ON (cb.ssd\_id = ssd.ssd\_id)

JOIN ram r ON (cb.ram\_id = r.ram\_id)

ORDER BY 1;

-- Query 3

-- This query displays the most expensive cpu.

SELECT TO\_CHAR(MAX(cost\_price), '$999.99') AS "Most expensive CPU"

FROM cpu

ORDER BY 1;

-- Query 4

-- This query displays all the builds purchased,

-- along with the build's purchase date, build id, and the name of the customer who purchased it.

SELECT first\_name || ' ' || last\_name AS "Customer Name", build\_id AS "Build purchased", purchase\_date AS "Purchase Date"

FROM customer c

JOIN builds\_purchased bp ON (c.customer\_id = bp.customer\_id)

ORDER BY 2;

-- Query 5

-- This query displays the customers who have purchased a build with a gpu that has a cost price

-- greater than the average cost price of all gpus, along with the associated build id, and the cost

-- price of the gpu associated with the build.

SELECT first\_name || ' ' || last\_name AS "Customer Name", bp.build\_id AS "Build ID", g.cost\_price AS "GPU Cost price"

FROM builds\_purchased bp

JOIN customer c ON (bp.customer\_id = c.customer\_id)

JOIN computer\_build cb ON (bp.build\_id = cb.build\_id)

JOIN gpu g ON (cb.gpu\_id = g.gpu\_id)

WHERE bp.build\_id IN

(SELECT build\_id FROM computer\_build cb

WHERE (SELECT g.cost\_price FROM gpu g WHERE g.gpu\_id = cb.gpu\_id)

> (SELECT AVG(cost\_price) FROM gpu))

ORDER BY 1;

-- Query 6

-- This query is a union of all brands and storages capacities of all hard drives and solid state drives,

-- along with a column specifying which type of drive it is.

SELECT hdd\_brand AS "Brand", hdd\_storage\_capacity AS "Storage Capacity", 'Hard Drive' AS "Type"

FROM hard\_drive

UNION

SELECT ssd\_brand, ssd\_storage\_capacity, 'Solid State Drive'

FROM solid\_state\_drive

ORDER BY 3;

-- Query 7

-- This query displays the years between the first recorded purchase date and the most recently recorded purchase date.

SELECT ROUND(MONTHS\_BETWEEN(MAX(purchase\_date), MIN(purchase\_date))/12, 1) AS "Years Between"

FROM builds\_purchased

ORDER BY 1;

-- Query 8

-- This query displays the storage capacities offered by the hard drives, converted into number format in order to properly

-- sort the storages from least to greatest (as integers, rather than strings).

SELECT DISTINCT TO\_NUMBER(REPLACE(hdd\_storage\_capacity, ' GB', '')) AS "Hard Drive Storages (in GB)"

FROM hard\_drive

ORDER BY 1;

-- Query 9

-- This query displays the customers who have purhcased computer build 3.

SELECT DISTINCT first\_name || ' ' || last\_name AS "Customer Name", street\_address AS "Street Address"

FROM customer c

JOIN builds\_purchased bp ON (c.customer\_id = bp.customer\_id)

WHERE (c.customer\_id, 3) IN (SELECT customer\_id, build\_id FROM builds\_purchased)

ORDER BY 1;

-- Query 10

-- This query displays the builds each customer has purchased (along with the purchase date), including the customers who have not purchased a build.

SELECT c.customer\_id, bp.build\_id, bp.purchase\_date

FROM customer c

FULL OUTER JOIN builds\_purchased bp ON (c.customer\_id = bp.customer\_id)

ORDER BY 1;

Part 5: Report

SET PAGESIZE 30

SET LINESIZE 70

SET FEEDBACK OFF

TTITLE 'Part 5 Semester Project|Each customer''s purchased builds'

BTITLE 'By - Robert Bryan'

BREAK ON customer\_id SKIP 2

COL customer\_id HEADING 'Customer ID' FORMAT 999

COL build\_id HEADING 'Build ID' FORMAT 999

COL purchase\_date HEADING 'Purchase Date' FORMAT A15

-- SQL Statement

-- Selects each customer that has purchased a build, what build id they have purchased, and the purchase date.

SELECT customer\_id, build\_id, purchase\_date FROM builds\_purchased

ORDER BY 1;

Part 5: Report Run

