

# CPSC 304 Project Cover Page

Milestone #: 4

Date: April 5, 2024

Group Number: 36

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Wendy Li	38036695	lwqz	wendyli.org@gmail.com
Antonia Tykei	82340886	atykei	antoniatykei@gmail.com
Wendy Tso	34159368	wendytso	wendywktso@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

## TABLE OF CONTENTS

<b>CPSC304 Project Repository: project_i0b5b_k9f0w_q9m1b</b>	<b>3</b>
<b>1. Project Summary</b>	<b>3</b>
<b>2. Notable Schema/Project Changes</b>	<b>3</b>
<b>3. SQL SCRIPT (DDL, Insert Statements and DROP Tables)</b>	<b>5</b>
<b>4. Schema and data after SQL initialization script</b>	<b>7</b>
• USER_NORMALIZED (UserID: varchar, email: varchar, experience level: varchar)	7
• EMAIL_DETERMINES_NAME (email: varchar, name: varchar)	9
• PLACES_ORDER (OrderID: varchar, UserID: varchar, date: date, total cost: real)	9
• KEYBOARD_CONTAINS (KeyboardName: varchar, OrderID: varchar, connection: varchar, date: date)	11
• MECHANICAL (KeyboardName: varchar, customizable actuation force : varchar)	12
• GAMING (KeyboardName: varchar, gaming software integration: varchar)	14
• KEYBOARD_ASSEMBLER_NORMALIZED (AssemblerID: int, experience level: varchar, number of completed keyboards: int)	16
• COMPLETE_EXPERIENCE_DETERMINES_SALARY (experience level, number of completed keyboards, salary)	17
• MADE_BY (AssemblerID: int, KeyboardName: varchar, date of completion: date)	18
• RATING_HAS_FEEDBACK (RatingID: varchar, KeyboardName: varchar, date: date)	20
• LIGHTING_NORMALIZED (LightingID: int, price: real, color: varchar, effect: varchar)	21
• COLOR_DETERMINES_BRIGHTNESS (color: varchar, brightness: int)	22
• CONFIGURES (LightingID: int, KeyboardName: varchar)	23
• ACCESSORY_INCLUDED_ON (AccessoryID: varchar, price: real, type: varchar, KeyboardName: varchar)	24
• BOARD_HAS_LAYOUT (BoardID: int, brand: varchar, material: varchar, price: real, size: real, type: varchar, KeyboardName: varchar)	25
• SWITCHES_ATTACH_TO (SwitchID: int, operating force: char[10], brand: varchar, type: varchar, price: real, lifespan: int, BoardID: int, KeyID: varchar)	26
• KEYCAP (KeyID: varchar, price: real, material: varchar, brand: varchar, SwitchID: int)	27
<b>5. Queries:</b>	<b>28</b>
a. INSERT	28
b. DELETE	31
c. UPDATE	33
d. SELECTION	36
e. PROJECTION	37
f. JOIN	38
g. AGGREGATION WITH GROUP BY	41
h. AGGREGATION WITH HAVING	42
i. NESTED AGGREGATION WITH GROUP BY	43
j. DIVISION	45
<b>6. README</b>	<b>47</b>

## CPSC304 Project Repository: [project\\_i0b5b\\_k9f0w\\_q9m1b](#)

[https://github.students.cs.ubc.ca/CPSC304-2023W-T2/project\\_i0b5b\\_k9f0w\\_q9m1b](https://github.students.cs.ubc.ca/CPSC304-2023W-T2/project_i0b5b_k9f0w_q9m1b)

### 1. Project Summary

The project focuses on mechanical keyboard customization, providing an informational database and ordering system. Users can design, organize, and place orders for customized keyboards through the platform, which aims to streamline the customization process.

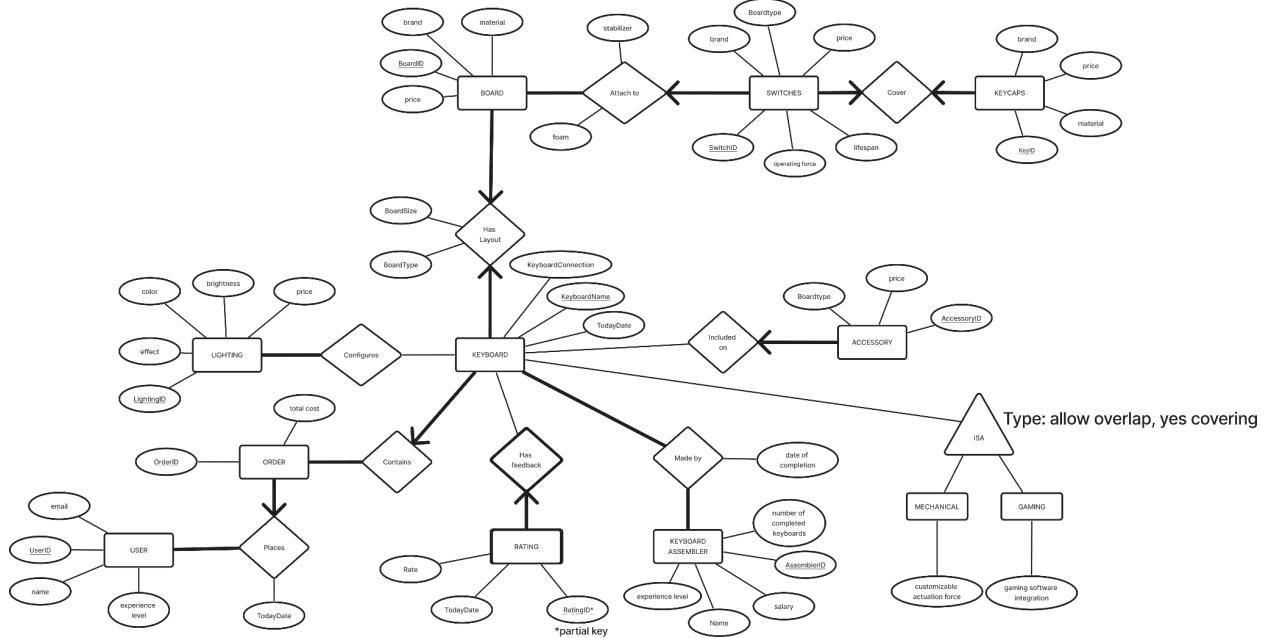
The keyboard application is a comprehensive keyboard customization database system designed for keyboard users and assemblers. It offers a range of functionalities to users, allowing them to manage orders, customize keyboards, analyze data, and interact with keyboard assemblers. Users can perform operations such as inserting new orders, updating existing orders, and deleting keyboards from the database. The application also supports more advanced operations like filtering users based on conditions, viewing specific data sets, and aggregating information for analysis. Users can also find keyboard assemblers based on their ratings, analyze salary summaries by experience level, and identify keyboards with specific configurations such as wave lighting effects. Overall, the keyboard application provides a simple interface for managing orders and exploring data related to keyboard customization.

### 2. Notable Schema/Project Changes

The updates to our schema were just the renaming of some attributes as we realized some of our names were reserved words in oracle SQL and although we could have used them - these changes were made to avoid any confusion.

Two new attributes were also added just to make some of our queries and results in the database more interesting and applicable. For example, we had a query to identify assemblers but just having their ID made the results not very interesting so we added the 'Name' attribute for our Assembler's too. Similarly, in our ratings, we added a numeric rating attribute ('Rate') that a user could provide feedback about their purchase (on a scale of 1-10). These changes are reflected in our updated schema with additional details below.

Updated ER diagram:



Updated schema:

- Renaming of attributes
- Additional attributes

1. EMAIL\_DETERMINES\_NAME (email: varchar, name: varchar)
2. USER\_NORMALIZED (UserID: varchar, email: varchar, experience\_level: varchar)
3. PLACES\_ORDER (OrderID: varchar, UserID: varchar, TodayDate: date, Total\_cost: real)
4. KEYBOARD\_CONTAINS (KeyboardName: varchar, OrderID: varchar, KeyboardConnection: varchar, TodayDate: date)
5. MECHANICAL (KeyboardName: varchar, Customizable\_actuation\_force: varchar)
6. GAMING (KeyboardName: varchar, Gaming\_software\_integration: varchar)
7. COMPLETE\_EXPERIENCE\_DETERMINES\_SALARY (Experience\_level, Number\_of\_completed\_keyboards, salary: real)
8. KEYBOARD\_ASSEMBLER\_NORMALIZED (AssemblerID: int, Name: varchar, Experience\_level: varchar, salary: real, Number\_of\_completed\_keyboards: int)

9. MADE\_BY (AssemblerID: int, **KeyboardName**: varchar, Date\_of\_completion: date)
10. RATING\_HAS\_FEEDBACK (RatingID: varchar, Rate: integer, **KeyboardName**: varchar, TodayDate: date)
11. LIGHTING\_NORMALIZED (LightingID: int, Price: real, **Color**: varchar, Effect: varchar)
12. COLOR\_DETERMINES\_BRIGHTNESS (Color: varchar, Brightness: int)
13. CONFIGURES (LightingID: int, **KeyboardName**: varchar)
14. ACCESSORY\_INCLUDED\_ON (AccessoryID: varchar, Price: real, **BoardType**: varchar, **KeyboardName**: varchar)
15. BOARD\_HAS\_LAYOUT (BoardID: int, Brand: varchar, MMaterial: varchar, Price: real, **BoardSize**: real, **BoardType**: varchar, **KeyboardName**: varchar)
16. SWITCHES\_ATTACH\_TO (SwitchID: int, Operating\_force: char[10], brand: varchar, **BoardType**: varchar, price: real, lifespan: int, **BoardID**: int, **KeyID**: varchar)
17. KEYCAP (KeyID: varchar, Price: real, Material: varchar, Brand: varchar, **SwitchID**: int)

### 3. SQL SCRIPT (DDL, Insert Statements and DROP Tables)

The Full SQL Script can be found here:

[https://github.students.cs.ubc.ca/CPSC304-2023W-T2/project\\_i0b5b\\_k9f0w\\_q9m1b/blob/main/Milestone4/setup.sql](https://github.students.cs.ubc.ca/CPSC304-2023W-T2/project_i0b5b_k9f0w_q9m1b/blob/main/Milestone4/setup.sql)

- **Drop Table** statements are lines 1-18 in the [setup.sql](#) file
- **Create Table** statements are on lines 20 - 158 in the [setup.sql](#) file
- **Insert Values** statements are on 161 - 484 in the [setup.sql](#) file

Below are some screenshots of the sql script

```
1  -- Drop Table Statements
2  DROP TABLE COLOR_DETERMINES_BRIGHTNESS;
3  DROP TABLE RATING_HAS_FEEDBACK;
4  DROP TABLE KEYCAP;
5  DROP TABLE SWITCHES_ATTACH_TO;
6  DROP TABLE BOARD_HAS_LAYOUT;
7  DROP TABLE ACCESSORY_INCLUDED_ON;
8  DROP TABLE CONFIGURES;
9  DROP TABLE LIGHTING_NORMALIZED;
10 DROP TABLE MADE_BY;
11 DROP TABLE GAMING;
12 DROP TABLE MECHANICAL;
13 DROP TABLE KEYBOARD_CONTAINS;
14 DROP TABLE PLACES_ORDER;
15 DROP TABLE USER_NORMALIZED;
16 DROP TABLE KEYBOARD_ASSEMBLER_NORMALIZED;
17 DROP TABLE COMPLETE_EXPERIENCE_DETERMINES_SALARY;
18 DROP TABLE EMAIL_DETERMINES_NAME;
19
20 -- CREATE TABLE Statements
21 CREATE TABLE EMAIL_DETERMINES_NAME (
22     Email VARCHAR(100) PRIMARY KEY,
23     Name VARCHAR(100)
24 );
25
26 CREATE TABLE USER_NORMALIZED (
27     UserID VARCHAR(100) PRIMARY KEY,
28     Email VARCHAR(100),
29     Experience_level VARCHAR(100) NOT NULL,
30     FOREIGN KEY (Email) REFERENCES EMAIL_DETERMINES_NAME (Email) ON DELETE SET NULL
31 );
32
```

```

136 CREATE TABLE SWITCHES_ATTACH_TO (
137     SwitchID INTEGER,
138     Operating_force VARCHAR(100),
139     Brand VARCHAR(100),
140     BoardType VARCHAR(100),
141     Price REAL,
142     Lifespan INTEGER,
143     BoardID INTEGER,
144     KeyID VARCHAR(100),
145     PRIMARY KEY (SwitchID),
146     FOREIGN KEY (BoardID) REFERENCES BOARD_HAS_LAYOUT(BoardID) ON DELETE CASCADE
147 );
148
149 CREATE TABLE KEYCAP (
150     KeyID VARCHAR(100) PRIMARY KEY,
151     Price REAL,
152     Material VARCHAR(100),
153     Brand VARCHAR(100),
154     SwitchID INTEGER UNIQUE,
155     FOREIGN KEY (SwitchID) REFERENCES SWITCHES_ATTACH_TO(SwitchID) ON DELETE SET NULL
156 );
157
158
159 -- Insert Statements
160 insert into color_determines_brightness (color, brightness) values ('RGB', 200000);
161 insert into color_determines_brightness (color, brightness) values ('Backlight', 300);
162 insert into color_determines_brightness (color, brightness) values ('White', 294849);
163 insert into color_determines_brightness (color, brightness) values ('Purple', 55);
164 insert into color_determines_brightness (color, brightness) values ('Blue', 71);
165
166 insert into email_determines_name (email, name) values ('ambikamod@hotmail.com', 'Ambika Mod');
167 insert into email_determines_name (email, name) values ('leo@gmail.com', 'Leo Woodall');
168 insert into email_determines_name (email, name) values ('ellie@gmail.com', 'Eleanor Tomlinson');
169 insert into email_determines_name (email, name) values ('jon@ubc.com', 'Jonny Weldon');
170 insert into email_determines_name (email, name) values ('essiedavis@gmail.com', 'Essie Davis');
171

```

#### 4. Schema and data after SQL initialization script

- **USER\_NORMALIZED (UserID: varchar, email: varchar, experience level: varchar)**
  - FK: *email* references EMAIL\_DETERMINES\_NAME

SQL> SELECT * FROM USER_NORMALIZED;		
USERID	EMAIL	EXPERIENCE_LEVEL
user1	ambikamod@hotmail.com	Intermediate
user2	leo@gmail.com	Advanced
user3	ellie@gmail.com	Beginner
user4	jon@ubc.com	Intermediate
user5	essiedavis@gmail.com	Advanced
user6	minag@yahoo.com	Beginner
user7	anniewang@gmail.com	Beginner
user8	jessiej@gmail.com	Advanced
user9	lawrenceholstaff@ubc.com	Intermediate
user10	donolo@gmail.com	Beginner

10 rows selected.

USERID	EMAIL	EXPERIENCE_LEVEL
user1	ambikamod@hotmail.com	Intermediate
user2	leo@gmail.com	Advanced
user3	ellie@gmail.com	Beginner
user4	jon@ubc.com	Intermediate
user5	essiedavis@gmail.com	Advanced
user6	minag@yahoo.com	Beginner
user7	anniewang@gmail.com	Beginner
user8	jessiej@gmail.com	Advanced
user9	lawrenceholstaff@ubc.com	Intermediate
user10	donolo@gmail.com	Beginner

---

- **EMAIL\_DETERMINES\_NAME** (email: varchar, name: varchar)

```
SQL> SELECT * FROM EMAIL_DETERMINES_NAME;
```

EMAIL	NAME
ambikamod@hotmail.com	Ambika Mod
leo@gmail.com	Leo Woodall
ellie@gmail.com	Eleanor Tomlinson
jon@ubc.com	Jonny Weldon
essiedavis@gmail.com	Essie Davis
minag@yahoo.com	Mina Gourdi
anniewang@gmail.com	Annie Wang
jessiej@gmail.com	Jessie Jay
lawrenceholstaff@ubc.com	Lawrence Holstaff
donolo@gmail.com	Donolo Miller

EMAIL	NAME
ambikamod@hotmail.com	Ambika Mod
leo@gmail.com	Leo Woodall
ellie@gmail.com	Eleanor Tomlinson
jon@ubc.com	Jonny Weldon
essiedavis@gmail.com	Essie Davis
minag@yahoo.com	Mina Gourdi
anniewang@gmail.com	Annie Wang
jessiej@gmail.com	Jessie Jay
lawrenceholstaff@ubc.com	Lawrence Holstaff
donolo@gmail.com	Donolo Miller

- **PLACES\_ORDER** (OrderID: varchar, UserID: varchar, date: date, total cost: real)

- *FK: UserID references USER*

```
SQL> SELECT * FROM PLACES_ORDER;
```

ORDERID	USERID	TODAYDATE	TOTAL_COST
order1	user1	19-MAR-24	150.99
order2	user2	18-MAR-24	299.99
order3	user3	17-MAR-24	1900.99
order4	user4	16-MAR-24	75.5
order5	user5	15-MAR-24	500
order6	user6	10-FEB-23	650.99
order7	user5	18-AUG-23	82.45
order8	user8	17-DEC-23	365.87
order9	user7	02-FEB-24	64.4
order10	user9	31-JAN-24	200
order11	user10	22-SEP-23	119.62
order12	user2	02-JAN-24	88.88
order13	user8	09-DEC-23	1000.5
order14	user9	01-APR-24	220.05
order15	user3	02-APR-24	103.72
order16	user3	31-JAN-24	200
order17	user3	22-SEP-23	119.62
order18	user3	02-JAN-24	88.88
order19	user8	09-DEC-23	1000.5
order20	user8	01-APR-24	220.05
order21	user8	02-APR-24	103.72
order22	user8	31-JAN-24	200
order23	user7	22-SEP-23	119.62
order24	user7	02-JAN-24	88.88
order25	user7	09-DEC-23	5000.5

<b>ORDERID</b>	<b>USERID</b>	<b>TODAYDATE</b>	<b>TOTAL_COST</b>
order1	user1	19-MAR-24	150.99
order2	user2	18-MAR-24	299.99
order3	user3	17-MAR-24	1900.99
order4	user4	16-MAR-24	75.5
order5	user5	15-MAR-24	500
order6	user6	10-FEB-23	650.99
order7	user5	18-AUG-23	82.45
order8	user8	17-DEC-23	365.87
order9	user7	02-FEB-24	64.4
order10	user9	31-JAN-24	200
order11	user10	22-SEP-23	119.62
order12	user2	02-JAN-24	88.88
order13	user8	09-DEC-23	1000.5
order14	user9	01-APR-24	220.05
order15	user3	02-APR-24	103.72
order16	user3	31-JAN-24	200
order17	user3	22-SEP-23	119.62
order18	user3	02-JAN-24	88.88
order19	user8	09-DEC-23	1000.5
order20	user8	01-APR-24	220.05
order21	user8	02-APR-24	103.72
order22	user8	31-JAN-24	200
order23	user7	22-SEP-23	119.62
order24	user7	02-JAN-24	88.88
order25	user7	09-DEC-23	5000.5

- KEYBOARD\_CONTAINS (KeyboardName: varchar, **OrderID**: varchar, connection: varchar, date: date)
  - *FK: OrderID references PLACES\_ORDER*

KEYBOARDNAME	ORDERID	KEYBOARDCONNECTION	TODAYDATE
keyboard1	order1	USB	19-MAR-24
keyboard2	order2	Wireless	18-MAR-24
keyboard3	order3	Bluetooth	17-MAR-24
keyboard4	order4	USB	16-MAR-24
keyboard5	order5	Wireless	15-MAR-24
keyboard6	order6	USB	10-FEB-23
keyboard7	order7	Wireless	18-AUG-23
keyboard8	order8	Bluetooth	17-DEC-23
keyboard9	order9	USB	02-FEB-24
keyboard10	order10	Wireless	31-JAN-24
keyboard11	order11	USB	22-SEP-23
keyboard12	order12	Bluetooth	02-JAN-24
keyboard13	order13	Bluetooth	09-DEC-23
keyboard14	order14	Wireless	01-APR-24
keyboard15	order15	Wireless	02-APR-24
keyboard16	order16	USB	10-FEB-23
keyboard17	order17	Wireless	18-AUG-23
keyboard18	order18	Bluetooth	17-DEC-23
keyboard19	order19	USB	02-FEB-24
keyboard20	order20	Wireless	31-JAN-24
keyboard21	order21	USB	22-SEP-23
keyboard22	order22	Bluetooth	02-JAN-24
keyboard23	order23	Bluetooth	09-DEC-23
keyboard24	order24	Wireless	01-APR-24
keyboard25	order25	Wireless	02-APR-24

KEYBOARDNAME	ORDERID	KEYBOARDCONNECTION	TODAYDATE
keyboard1	order1	USB	19-MAR-24
keyboard2	order2	Wireless	18-MAR-24
keyboard3	order3	Bluetooth	17-MAR-24
keyboard4	order4	USB	16-MAR-24
keyboard5	order5	Wireless	15-MAR-24
keyboard6	order6	USB	10-FEB-23
keyboard7	order7	Wireless	18-AUG-23
keyboard8	order8	Bluetooth	17-DEC-23
keyboard9	order9	USB	02-FEB-24
keyboard10	order10	Wireless	31-JAN-24
keyboard11	order11	USB	22-SEP-23
keyboard12	order12	Bluetooth	02-JAN-24
keyboard13	order13	Bluetooth	09-DEC-23
keyboard14	order14	Wireless	01-APR-24
keyboard15	order15	Wireless	02-APR-24
keyboard16	order16	USB	10-FEB-23
keyboard17	order17	Wireless	18-AUG-23
keyboard18	order18	Bluetooth	17-DEC-23
keyboard19	order19	USB	02-FEB-24
keyboard20	order20	Wireless	31-JAN-24
keyboard21	order21	USB	22-SEP-23
keyboard22	order22	Bluetooth	02-JAN-24
keyboard23	order23	Bluetooth	09-DEC-23
keyboard24	order24	Wireless	01-APR-24
keyboard25	order25	Wireless	02-APR-24

- MECHANICAL (KeyboardName: varchar, customizable actuation force : varchar)

- FK: *KeyboardName* references ('superclass') KEYBOARD\_CONTAINS

KEYBOARDNAME	CUSTOMIZABLE_ACTUATION_FORCE
keyboard1	60g
keyboard2	65g
keyboard3	55g
keyboard4	70g
keyboard5	45g
keyboard6	60g
keyboard7	65g
keyboard8	55g
keyboard9	70g
keyboard10	45g
keyboard11	60g
keyboard12	65g
keyboard13	55g
keyboard14	70g
keyboard15	45g
keyboard16	60g
keyboard17	65g
keyboard18	55g
keyboard19	70g
keyboard20	45g
keyboard21	60g
keyboard22	65g
keyboard23	55g
keyboard24	70g
keyboard25	45g

KEYBOARDNAME	CUSTOMIZABLE_ACTUATION_FORCE
keyboard1	60g
keyboard2	65g
keyboard3	55g
keyboard4	70g
keyboard5	45g
keyboard6	60g
keyboard7	65g
keyboard8	55g
keyboard9	70g
keyboard10	45g
keyboard11	60g
keyboard12	65g
keyboard13	55g
keyboard14	70g
keyboard15	45g
keyboard16	60g
keyboard17	65g
keyboard18	55g
keyboard19	70g
keyboard20	45g
keyboard21	60g
keyboard22	65g
keyboard23	55g
keyboard24	70g
keyboard25	45g

- **GAMING (KeyboardName: varchar, gaming software integration: varchar)**

- *FK: KeyboardName references ('superclass') KEYBOARD\_CONTAINS*

KEYBOARDNAME	GAMING_SOFTWARE_INTEGRATION
keyboard1	Razer Synapse
keyboard2	Logitech G HUB
keyboard3	Corsair iCUE
keyboard4	SteelSeries Engine
keyboard5	ASUS Armoury Crate
keyboard6	Razer Synapse
keyboard7	Logitech G HUB
keyboard8	Corsair iCUE
keyboard9	SteelSeries Engine
keyboard10	ASUS Armoury Crate
keyboard11	Razer Synapse
keyboard12	Logitech G HUB
keyboard13	Corsair iCUE
keyboard14	SteelSeries Engine
keyboard15	ASUS Armoury Crate
keyboard16	Razer Synapse
keyboard17	Logitech G HUB
keyboard18	Corsair iCUE
keyboard19	SteelSeries Engine
keyboard20	ASUS Armoury Crate
keyboard21	Razer Synapse
keyboard22	Logitech G HUB
keyboard23	Corsair iCUE
keyboard24	SteelSeries Engine
keyboard25	ASUS Armoury Crate

KEYBOARDNAME	GAMING_SOFTWARE_INTEGRATION
keyboard1	Razer Synapse
keyboard2	Logitech G HUB
keyboard3	Corsair iCUE
keyboard4	SteelSeries Engine
keyboard5	ASUS Armoury Crate
keyboard6	Razer Synapse
keyboard7	Logitech G HUB
keyboard8	Corsair iCUE
keyboard9	SteelSeries Engine
keyboard10	ASUS Armoury Crate
keyboard11	Razer Synapse
keyboard12	Logitech G HUB
keyboard13	Corsair iCUE
keyboard14	SteelSeries Engine
keyboard15	ASUS Armoury Crate
keyboard16	Razer Synapse
keyboard17	Logitech G HUB
keyboard18	Corsair iCUE
keyboard19	SteelSeries Engine
keyboard20	ASUS Armoury Crate
keyboard21	Razer Synapse
keyboard22	Logitech G HUB
keyboard23	Corsair iCUE
keyboard24	SteelSeries Engine
keyboard25	ASUS Armoury Crate

- **KEYBOARD\_ASSEMBLER\_NORMALIZED (AssemblerID: int, experience level: varchar, number of completed keyboards: int)**
  - *FK: experience level references COMPLETE\_EXPERIENCE\_DETERMINES\_SALARY*
  - *FK: number of completed keyboards references COMPLETE\_EXPERIENCE\_DETERMINES\_SALARY*

SQL> SELECT * FROM KEYBOARD_ASSEMBLER_NORMALIZED;			
ASSEMBLERIDNAME	EXPERIENCE_LEVEL	SALARY	NUMBER_OF_COMPLETED_KEYBOARDS
1Thelonius Ellison	Beginner	800	10
2Sinatra Golden	Intermediate	2400	20
3Carl Brunt	Advanced	3200	30
4Aileen Hoover	Intermediate	2500	25
5Coraline Alexander	Advanced	3600	35
6George Mcyntire	Beginner	1800	8
7Vivian Lee	Intermediate	3400	24
8Carl Brunt	Advanced	4000	50
9Aileen Hoover	Intermediate	3100	30
10Roger Ted	Advanced	3650	30

10 rows selected.

ASSEMBLERID	NAME	EXPERIENCE_LEVEL	SALARY	NUMBER_OF_COMPLETED_KEYBOARDS
1	Thelonius Ellison	Beginner	800	10
2	Sinatra Golden	Intermediate	2400	20
3	Carl Brunt	Advanced	3200	30
4	Aileen Hoover	Intermediate	2500	25
5	Coraline Alexander	Advanced	3600	35
6	George Mcyntire	Beginner	1800	8
7	Vivian Lee	Intermediate	3400	24
8	Carl Brunt	Advanced	4000	50
9	Aileen Hoover	Intermediate	3100	30
10	Roger Ted	Advanced	3650	30

- **COMPLETE\_EXPERIENCE\_DETERMINES\_SALARY** (experience level, number of completed keyboards, salary)

```
SQL> SELECT * FROM COMPLETE_EXPERIENCE_DETERMINES_SALARY;

EXPERIENCE_LEVEL          SALARY NUMBER_OF_COMPLETED_KEYBOARDS
-----          -----
Beginner                  800      10
Intermediate              2400     20
Advanced                  3200     30
Intermediate              2500     25
Advanced                  3600     35
Beginner                  1800      8
Intermediate              3400     24
Intermediate              3200     30
Beginner                  500      5
Advanced                  4000     50

10 rows selected.
```

EXPERIENCE_LEVEL	SALARY	NUMBER_OF_COMPLETED_KEYBOARDS
Beginner	800	10
Intermediate	2400	20
Advanced	3200	30
Intermediate	2500	25
Advanced	3600	35
Beginner	1800	8
Intermediate	3400	24
Intermediate	3200	30
Beginner	500	5
Advanced	4000	50

- 
- **MADE\_BY (AssemblerID: int, KeyboardName: varchar, date of completion: date)**

```
SQL> SELECT * FROM MADE_BY;

ASSEMBLER ID KEYBOARD NAME          DATE OF COMPLET
-----  -----  -----
1keyboard1           19-MAR-24
2keyboard2           18-MAR-24
3keyboard3           17-MAR-24
4keyboard4           20-MAR-24
5keyboard5           25-MAR-24
1keyboard6           26-FEB-24
2keyboard7           10-SEP-23
3keyboard8           23-DEC-23
4keyboard9           19-FEB-24
5keyboard10          07-MAR-24
1keyboard11          11-OCT-23
2keyboard12          17-JAN-24
3keyboard13          17-DEC-23
4keyboard14          13-APR-24
5keyboard15          13-APR-24
1keyboard16          26-FEB-24
2keyboard17          10-SEP-23
3keyboard18          23-DEC-23
4keyboard19          19-FEB-24
5keyboard20          07-MAR-24
9keyboard21          11-OCT-23
2keyboard22          17-JAN-24
7keyboard23          17-DEC-23
4keyboard24          13-APR-24
6keyboard25          13-APR-24

25 rows selected.
```

<b>ASSEMBLERID</b>	<b>KEYBOARDNAME</b>	<b>DATE_OF_COMPLETION</b>
1	keyboard1	19-MAR-24
2	keyboard2	18-MAR-24
3	keyboard3	17-MAR-24
4	keyboard4	20-MAR-24
5	keyboard5	25-MAR-24
1	keyboard6	26-FEB-24
2	keyboard7	10-SEP-23
3	keyboard8	23-DEC-23
4	keyboard9	19-FEB-24
5	keyboard10	07-MAR-24
1	keyboard11	11-OCT-23
2	keyboard12	17-JAN-24
3	keyboard13	17-DEC-23
4	keyboard14	13-APR-24
5	keyboard15	13-APR-24
1	keyboard16	26-FEB-24
2	keyboard17	10-SEP-23
3	keyboard18	23-DEC-23
4	keyboard19	19-FEB-24
5	keyboard20	07-MAR-24
9	keyboard21	11-OCT-23
2	keyboard22	17-JAN-24
7	keyboard23	17-DEC-23
4	keyboard24	13-APR-24
6	keyboard25	13-APR-24

---

- **RATING\_HAS\_FEEDBACK (RatingID: varchar, KeyboardName: varchar, date: date)**

RATINGID	RATEKEYBOARDNAME	TODAYDATE
rating1	9keyboard1	19-MAR-24
rating2	8keyboard2	18-MAR-24
rating3	1keyboard3	17-MAR-24
rating4	2keyboard4	16-MAR-24
rating5	3keyboard5	15-MAR-24
rating6	8keyboard6	10-FEB-23
rating7	9keyboard7	18-AUG-23
rating8	10keyboard8	17-DEC-23
rating9	4keyboard9	02-FEB-24
rating10	5keyboard10	31-JAN-24
rating11	6keyboard11	22-SEP-23
rating12	7keyboard12	02-JAN-24
rating13	8keyboard13	09-DEC-23
rating14	3keyboard14	01-APR-24
rating15	2keyboard15	02-APR-24
rating16	2keyboard16	10-FEB-23
rating17	4keyboard17	18-AUG-23
rating18	9keyboard18	17-DEC-23
rating19	7keyboard19	02-FEB-24
rating20	6keyboard20	31-JAN-24
rating21	5keyboard21	22-SEP-23
rating22	5keyboard22	02-JAN-24
rating23	8keyboard23	09-DEC-23
rating24	8keyboard24	01-APR-24
rating25	4keyboard25	02-APR-24

25 rows selected.

RATINGID	RATE	KEYBOARDNAME	TODAYDATE
rating1	9	keyboard1	19-MAR-24
rating2	8	keyboard2	18-MAR-24
rating3	1	keyboard3	17-MAR-24
rating4	2	keyboard4	16-MAR-24
rating5	3	keyboard5	15-MAR-24
rating6	8	keyboard6	10-FEB-23
rating7	9	keyboard7	18-AUG-23
rating8	10	keyboard8	17-DEC-23
rating9	4	keyboard9	02-FEB-24
rating10	5	keyboard10	31-JAN-24
rating11	6	keyboard11	22-SEP-23
rating12	7	keyboard12	02-JAN-24
rating13	8	keyboard13	09-DEC-23
rating14	3	keyboard14	01-APR-24
rating15	2	keyboard15	02-APR-24
rating16	2	keyboard16	10-FEB-23
rating17	4	keyboard17	18-AUG-23
rating18	9	keyboard18	17-DEC-23
rating19	7	keyboard19	02-FEB-24
rating20	6	keyboard20	31-JAN-24
rating21	5	keyboard21	22-SEP-23
rating22	5	keyboard22	02-JAN-24
rating23	8	keyboard23	09-DEC-23
rating24	8	keyboard24	01-APR-24
rating25	4	keyboard25	02-APR-24

- **LIGHTING\_NORMALIZED (LightingID: int, price: real, color: varchar, effect: varchar)**
  - *FK: color references COLOR\_DETERMINES\_BRIGHTNESS, color is NOT NULL*

```
SQL> SELECT * FROM LIGHTING_NORMALIZED;

LIGHTINGID PRICECOLOR                                     EFFECT
-----
1 49.99RGB                                         Wave
2 29.99White                                        Wave
3 39.99Blue                                         Wave
4 19.99Purple                                        Wave
5 59.99RGB                                         Rainbow
6 49.99RGB                                         Breathing
7 29.99White                                        Cycle
8 39.99Blue                                         Cycle
9 19.99Purple                                        Cycle

9 rows selected.
```

LIGHTINGID	PRICE	COLOR	EFFECT
1	49.99	RGB	Wave
2	29.99	White	Wave
3	39.99	Blue	Wave
4	19.99	Purple	Wave
5	59.99	RGB	Rainbow
6	49.99	RGB	Breathing
7	29.99	White	Cycle
8	39.99	Blue	Cycle
9	19.99	Purple	Cycle

- COLOR\_DETERMINES\_BRIGHTNESS (color: varchar, brightness: int)

```
SQL> SELECT * FROM COLOR_DETERMINES_BRIGHTNESS;

COLOR                                BRIGHTNESS
-----
RGB                                    200000
Backlight                             300
White                                  294849
Purple                                 55
Blue                                   71
```

COLOR	BRIGHTNESS
RGB	200000
Backlight	300
White	294849
Purple	55
Blue	71

- **CONFIGURES** (LightingID: int, KeyboardName: varchar)

- FK: *KeyboardName* references KEYBOARD\_CONTAINS
- FK: *LightingID* references LIGHTING

```
SQL> SELECT * FROM CONFIGURES;
          LIGHTING ID KEYBOARD NAME
-----+
1keyboard1
1keyboard10
1keyboard11
1keyboard15
1keyboard19
2keyboard12
2keyboard16
2keyboard2
2keyboard20
3keyboard13
3keyboard17
3keyboard21
3keyboard3
4keyboard14
4keyboard18
4keyboard22
4keyboard4
5keyboard5
6keyboard6
7keyboard23
7keyboard7
8keyboard24
8keyboard8
9keyboard25
9keyboard9
25 rows selected.
```

LIGHTINGID	KEYBOARDNAME
1	keyboard1
1	keyboard10
1	keyboard11
1	keyboard15
1	keyboard19
2	keyboard12
2	keyboard16
2	keyboard2
2	keyboard20
3	keyboard13
3	keyboard17
3	keyboard21
3	keyboard3
4	keyboard14
4	keyboard18
4	keyboard22
4	keyboard4
5	keyboard5
6	keyboard6
7	keyboard23
7	keyboard7
8	keyboard24
8	keyboard8
9	keyboard25
9	keyboard9

- **ACCESSORY\_INCLUDED\_ON** (AccessoryID: varchar, price: real, type: varchar, KeyboardName: varchar)
  - FK: *KeyboardName* references *KEYBOARD\_CONTAINS*

```
SQL> SELECT * FROM ACCESSORY_INCLUDED_ON;
```

ACCESSORYID	Price	BOARDTYPE	KEYBOARDNAME
acc1	\$19.99	Full-size	keyboard1
acc2	\$14.99	Tenkeyless	keyboard2
acc3	\$9.99	Compact	keyboard3
acc4	\$24.99	Full-size	keyboard4
acc5	\$29.99	Tenkeyless	keyboard5
acc6	\$19.99	Full-size	keyboard6
acc7	\$14.99	Tenkeyless	keyboard7
acc8	\$9.99	Compact	keyboard8
acc9	\$24.99	Full-size	keyboard9
acc10	\$29.99	Tenkeyless	keyboard10
acc11	\$19.99	Full-size	keyboard11
acc12	\$14.99	Tenkeyless	keyboard12
acc13	\$9.99	Compact	keyboard13
acc14	\$24.99	Full-size	keyboard14
acc15	\$29.99	Tenkeyless	keyboard15

15 rows selected.

ACCESSORYID	PRICE	BOARDTYPE	KEYBOARDNAME
acc1	19.99	Full-size	keyboard1
acc2	14.99	Tenkeyless	keyboard2
acc3	9.99	Compact	keyboard3
acc4	24.99	Full-size	keyboard4
acc5	29.99	Tenkeyless	keyboard5
acc6	19.99	Full-size	keyboard6
acc7	14.99	Tenkeyless	keyboard7
acc8	9.99	Compact	keyboard8
acc9	24.99	Full-size	keyboard9
acc10	29.99	Tenkeyless	keyboard10
acc11	19.99	Full-size	keyboard11
acc12	14.99	Tenkeyless	keyboard12
acc13	9.99	Compact	keyboard13
acc14	24.99	Full-size	keyboard14
acc15	29.99	Tenkeyless	keyboard15

- **BOARD\_HAS\_LAYOUT** (BoardID: int, brand: varchar, material: varchar, price: real, size: real, type: varchar, KeyboardName: varchar)

- *FK: KeyboardName UNIQUE (one to one - so it is CK)*

```
SQL> SELECT * FROM BOARD_HAS_LAYOUT;
rows will be truncated

BOARDID BRAND MATERIAL PRICE BOARD SIZE BOARDTYPE
-----  -----
1Logitech Plastic 99.99 17.3Full-size
2Corsair Aluminum 149.99 15.6Tenkeyless
3Razer Plastic 79.99 14Compact
4SteelSeries Aluminum 129.99 17.3Full-size
5Ducky Plastic 109.99 15.6Tenkeyless
6Logitech Plastic 99.99 17.3Full-size
7Corsair Aluminum 149.99 15.6Tenkeyless
8Razer Plastic 79.99 14Compact
9SteelSeries Aluminum 129.99 17.3Full-size
10Ducky Plastic 109.99 15.6Tenkeyless
11Logitech Plastic 99.99 17.3Full-size
12Corsair Aluminum 149.99 15.6Tenkeyless
13Razer Plastic 79.99 14Compact
14SteelSeries Aluminum 129.99 17.3Full-size
15Ducky Plastic 109.99 15.6Tenkeyless
16Logitech Plastic 99.99 17.3Full-size
17Corsair Plastic 99.99 17.3Full-size
18Razer Aluminum 149.99 15.6Tenkeyless
19SteelSeries Plastic 79.99 14Compact
20Ducky Aluminum 129.99 17.3Full-size
21Logitech Plastic 109.99 15.6Tenkeyless
22Corsair Plastic 99.99 17.3Full-size
23Razer Aluminum 149.99 15.6Tenkeyless
24SteelSeries Plastic 79.99 14Compact
25Ducky Aluminum 129.99 17.3Full-size
25Ducky Plastic 109.99 15.6Tenkeyless

25 rows selected.
```

BOARDID	BRAND	MATERIAL	PRICE	BOARDSIZE	BOARDDTYPE	KEYBOARDNAME
1	Logitech	Plastic	99.99	17.3	Full-size	keyboard1
2	Corsair	Aluminum	149.99	15.6	Tenkeyless	keyboard2
3	Razer	Plastic	79.99	14	Compact	keyboard3
4	SteelSeries	Aluminum	129.99	17.3	Full-size	keyboard4
5	Ducky	Plastic	109.99	15.6	Tenkeyless	keyboard5
6	Logitech	Plastic	99.99	17.3	Full-size	keyboard6
7	Corsair	Aluminum	149.99	15.6	Tenkeyless	keyboard7
8	Razer	Plastic	79.99	14	Compact	keyboard8
9	SteelSeries	Aluminum	129.99	17.3	Full-size	keyboard9
10	Ducky	Plastic	109.99	15.6	Tenkeyless	keyboard10
11	Logitech	Plastic	99.99	17.3	Full-size	keyboard11
12	Corsair	Aluminum	149.99	15.6	Tenkeyless	keyboard12
13	Razer	Plastic	79.99	14	Compact	keyboard13
14	SteelSeries	Aluminum	129.99	17.3	Full-size	keyboard14
15	Ducky	Plastic	109.99	15.6	Tenkeyless	keyboard15
16	Logitech	Plastic	99.99	17.3	Full-size	keyboard16
17	Corsair	Aluminum	149.99	15.6	Tenkeyless	keyboard17
18	Razer	Plastic	79.99	14	Compact	keyboard18
19	SteelSeries	Aluminum	129.99	17.3	Full-size	keyboard19
20	Ducky	Plastic	109.99	15.6	Tenkeyless	keyboard20
21	Logitech	Plastic	99.99	17.3	Full-size	keyboard21
22	Corsair	Aluminum	149.99	15.6	Tenkeyless	keyboard22
23	Razer	Plastic	79.99	14	Compact	keyboard23
24	SteelSeries	Aluminum	129.99	17.3	Full-size	keyboard24
25	Ducky	Plastic	109.99	15.6	Tenkeyless	keyboard25

- 
- **SWITCHES\_ATTACH\_TO (SwitchID: int, operating force: char[10], brand: varchar, type: varchar, price: real, lifespan: int, BoardID: int, KeyID: varchar)**
    - *FK: BoardID references BOARD\_HAS\_LAYOUT, is NOT NULL*
    - *FK: KeyID references KEYCAP*

SQL> SELECT * FROM SWITCHES_ATTACH_TO;								
SWITCH_ID	OPERATING_FORCE	BRAND	BOARD_TYPE	PRICE	LIFESPAN	BOARD_ID	KEY_ID	ID
	150g	Cherry	Full-size	5.99	5000000	1	key1	
	260g	Razer	Tenkeyless	6.99	6000000	2	key2	
	370g	Logitech	Compact	7.99	7000000	3	key3	
	480g	SteelSeries	Full-size	8.99	8000000	4	key4	
	590g	Ducky	Tenkeyless	9.99	9000000	5	key5	
	650g	Cherry	Full-size	5.99	5000000	6	key6	
	760g	Razer	Tenkeyless	6.99	6000000	7	key7	
	870g	Logitech	Compact	7.99	7000000	8	key8	
	980g	SteelSeries	Full-size	8.99	8000000	9	key9	
	1090g	Ducky	Tenkeyless	9.99	9000000	10	key10	
	1150g	Cherry	Full-size	5.99	5000000	11	key11	
	1260g	Razer	Tenkeyless	6.99	6000000	12	key12	
	1370g	Logitech	Compact	7.99	7000000	13	key13	
	1480g	SteelSeries	Full-size	8.99	8000000	14	key14	
	1590g	Ducky	Tenkeyless	9.99	9000000	15	key15	

15 rows selected.

SWITCHID	OPERATING_FORCE	BRAND	BOARDDTYPE	PRICE	LIFESPAN	BOARDID	KEYID
1	50g	Cherry	Full-size	5.99	5000000	1	key1
2	60g	Razer	Tenkeyless	6.99	6000000	2	key2
3	70g	Logitech	Compact	7.99	7000000	3	key3
4	80g	SteelSeries	Full-size	8.99	8000000	4	key4
5	90g	Ducky	Tenkeyless	9.99	9000000	5	key5
6	50g	Cherry	Full-size	5.99	5000000	6	key6
7	60g	Razer	Tenkeyless	6.99	6000000	7	key7
8	70g	Logitech	Compact	7.99	7000000	8	key8
9	80g	SteelSeries	Full-size	8.99	8000000	9	key9
10	90g	Ducky	Tenkeyless	9.99	9000000	10	key10
11	50g	Cherry	Full-size	5.99	5000000	11	key11
12	60g	Razer	Tenkeyless	6.99	6000000	12	key12
13	70g	Logitech	Compact	7.99	7000000	13	key13
14	80g	SteelSeries	Full-size	8.99	8000000	14	key14
15	90g	Ducky	Tenkeyless	9.99	9000000	15	key15

- **KEYCAP (KeyID: varchar, price: real, material: varchar, brand: varchar, SwitchID: int)**
  - *SwitchID is UNIQUE, is NOT NULL*

```
SQL> SELECT * FROM KEYCAP;

```

KEYID	PRICEMATERIAL	BRAND	SWITCHID
keycap1	2.99ABS	Generic	1
keycap2	3.99PBT	Corsair	2
keycap3	4.99ABS	Razer	3
keycap4	5.99PBT	SteelSeries	4
keycap5	6.99ABS	Ducky	5
keycap6	2.99ABS	Generic	6
keycap7	3.99PBT	Generic	7
keycap8	4.99ABS	Razer	8
keycap9	5.99PBT	Ducky	9
keycap10	6.99ABS	Ducky	10
keycap11	2.99ABS	Generic	11
keycap12	3.99PBT	Ducky	12
keycap13	4.99ABS	Razer	13
keycap14	5.99PBT	Generic	14
keycap15	6.99ABS	Ducky	15

15 rows selected.

KEYID	PRICE	MATERIAL	BRAND	SWITCHID
keycap1	2.99	ABS	Generic	1
keycap2	3.99	PBT	Corsair	2
keycap3	4.99	ABS	Razer	3
keycap4	5.99	PBT	SteelSeries	4
keycap5	6.99	ABS	Ducky	5
keycap6	2.99	ABS	Generic	6
keycap7	3.99	PBT	Generic	7
keycap8	4.99	ABS	Razer	8
keycap9	5.99	PBT	Ducky	9
keycap10	6.99	ABS	Ducky	10
keycap11	2.99	ABS	Generic	11
keycap12	3.99	PBT	Ducky	12
keycap13	4.99	ABS	Razer	13
keycap14	5.99	PBT	Generic	14
keycap15	6.99	ABS	Ducky	15

## 5. Queries:

### a. INSERT

- FileName + Link: [INSERT\\_order.php](#)
- Line Number Reference for Implementation: 105
- SQL command in our code:

```
insert into places_order values (:bind1, :bind2, :bind3, :bind4)
```

- Function with the above code: handleInsertRequest (lines 35 - 122)

**Before Screenshot:****Insert New Order for Existing User**

Inserts new order information for existing userID

OrderId:

UserId:

Date:

Cost:

**Insert**

ORDERID	USERID	TODAYDATE	TOTAL_COST
order1	user1	19-MAR-24	150.99
order2	user2	18-MAR-24	299.99
order3	user3	17-MAR-24	1900.99
order4	user4	16-MAR-24	75.5
order5	user5	15-MAR-24	500
order6	user6	10-FEB-23	650.99
order7	user5	18-AUG-23	82.45
order8	user8	17-DEC-23	365.87
order9	user7	02-FEB-24	64.4
order10	user9	31-JAN-24	200
order11	user10	22-SEP-23	119.62
order12	user2	02-JAN-24	88.88
order13	user8	09-DEC-23	1000.5
order14	user9	01-APR-24	220.05
order15	user3	02-APR-24	103.72
order16	user3	31-JAN-24	200
order17	user3	22-SEP-23	119.62
order18	user3	02-JAN-24	88.88
order19	user8	09-DEC-23	1000.5
order20	user8	01-APR-24	220.05
order21	user8	02-APR-24	103.72
order22	user8	31-JAN-24	200
order23	user7	22-SEP-23	119.62
order24	user7	02-JAN-24	88.88
order25	user7	09-DEC-23	5000.5

**During Screenshot:**

## Insert New Order for Existing User

Inserts new order information for existing userID

OrderId:

UserId:

Date:

Cost:

[View All Tables](#) [Go Back to Main Page](#)

### After Screenshot:

Insertion successful!			
ORDERID	USERID	TODAYDATE	TOTAL_COST
order1	user1	19-MAR-24	150.99
order2	user2	18-MAR-24	299.99
order3	user3	17-MAR-24	1900.99
order4	user4	16-MAR-24	75.5
order5	user5	15-MAR-24	500
order6	user6	10-FEB-23	650.99
order7	user5	18-AUG-23	82.45
order8	user8	17-DEC-23	365.87
order9	user7	02-FEB-24	64.4
order10	user9	31-JAN-24	200
order11	user10	22-SEP-23	119.62
order12	user2	02-JAN-24	88.88
order13	user8	09-DEC-23	1000.5
order14	user9	01-APR-24	220.05
order15	user3	02-APR-24	103.72
order16	user3	31-JAN-24	200
order17	user3	22-SEP-23	119.62
order18	user3	02-JAN-24	88.88
order19	user8	09-DEC-23	1000.5
order20	user8	01-APR-24	220.05
order21	user8	02-APR-24	103.72
order22	user8	31-JAN-24	200
order23	user7	22-SEP-23	119.62
order24	user7	02-JAN-24	88.88
order25	user7	09-DEC-23	5000.5
Aorder40	user1	01-JAN-24	1000

**b. DELETE**

- **FileName + Link:** [DEL\\_keyboard.php](#)
- **Line Number Reference for Implementation:** 65
- **SQL command in our code:**

```
delete from KEYBOARD_CONTAINS where KeyboardName = (:bind1)
```

- **Function with the above code:** handleDeleteKeyboardReq (lines 36 - 107)

**Before Screenshot:**

Below is the KEYBOARD\_CONTAINS table

KEYBOARDNAME	ORDERID	KEYBOARDCONNECTION	TODAYDATE
keyboard1	order1	USB	19-MAR-24
keyboard2	order2	Wireless	18-MAR-24
keyboard3	order3	Bluetooth	17-MAR-24
keyboard4	order4	USB	16-MAR-24
keyboard5	order5	Wireless	15-MAR-24
keyboard6	order6	USB	10-FEB-23
keyboard7	order7	Wireless	18-AUG-23
keyboard8	order8	Bluetooth	17-DEC-23
keyboard9	order9	USB	02-FEB-24
keyboard10	order10	Wireless	31-JAN-24
keyboard11	order11	USB	22-SEP-23
keyboard12	order12	Bluetooth	02-JAN-24
keyboard13	order13	Bluetooth	09-DEC-23
keyboard14	order14	Wireless	01-APR-24
keyboard15	order15	Wireless	02-APR-24
keyboard16	order16	USB	10-FEB-23
keyboard17	order17	Wireless	18-AUG-23
keyboard18	order18	Bluetooth	17-DEC-23
keyboard19	order19	USB	02-FEB-24
keyboard20	order20	Wireless	31-JAN-24
keyboard21	order21	USB	22-SEP-23
keyboard22	order22	Bluetooth	02-JAN-24
keyboard23	order23	Bluetooth	09-DEC-23
keyboard24	order24	Wireless	01-APR-24
keyboard25	order25	Wireless	02-APR-24

**During Screenshot:**

## Delete Keyboard

Please indicate the KeyboardName you wish to delete.

KeyboardName:

[View All Tables](#) [Go Back to Main Page](#)

**After Screenshot:**

Since KeyboardName is a Foreign Key in other tables, we can see it is from the example that keyboard3 was removed from the main KEYBOARD\_CONTAINS table and also all additional tables that have keyboardname is a foreign key (due to the delete cascade restriction)

Deletion successful!

KEYBOARDNAME	ORDERID	KEYBOARD CONNECTION	TODAYDATE
keyboard1	order1	USB	19-MAR-24
keyboard2	order2	Wireless	18-MAR-24
keyboard4	order4	USB	16-MAR-24
keyboard5	order5	Wireless	15-MAR-24
keyboard6	order6	USB	10-FEB-23
keyboard7	order7	Wireless	18-AUG-23
keyboard8	order8	Bluetooth	17-DEC-23
keyboard9	order9	USB	02-FEB-24
keyboard10	order10	Wireless	31-JAN-24
keyboard11	order11	USB	22-SEP-23
keyboard12	order12	Bluetooth	02-JAN-24
keyboard13	order13	Bluetooth	09-DEC-23
keyboard14	order14	Wireless	01-APR-24
keyboard15	order15	Wireless	02-APR-24
keyboard16	order16	USB	10-FEB-23
keyboard17	order17	Wireless	18-AUG-23
keyboard18	order18	Bluetooth	17-DEC-23
keyboard19	order19	USB	02-FEB-24
keyboard20	order20	Wireless	31-JAN-24
keyboard21	order21	USB	22-SEP-23
keyboard22	order22	Bluetooth	02-JAN-24
keyboard23	order23	Bluetooth	09-DEC-23
keyboard24	order24	Wireless	01-APR-24
keyboard25	order25	Wireless	02-APR-24

ASSEMBLERID	KEYBOARDNAME	DATE_OF_COMPLETION
1	keyboard1	19-MAR-24
2	keyboard2	18-MAR-24
4	keyboard4	20-MAR-24
5	keyboard5	25-MAR-24
1	keyboard6	26-FEB-24
2	keyboard7	10-SEP-23
3	keyboard8	23-DEC-23
4	keyboard9	19-FEB-24
5	keyboard10	07-MAR-24
1	keyboard11	11-OCT-23
2	keyboard12	17-JAN-24
3	keyboard13	17-DEC-23
4	keyboard14	13-APR-24
5	keyboard15	13-APR-24
1	keyboard16	26-FEB-24
2	keyboard17	10-SEP-23
3	keyboard18	23-DEC-23
4	keyboard19	19-FEB-24
5	keyboard20	07-MAR-24
9	keyboard21	11-OCT-23
2	keyboard22	17-JAN-24
7	keyboard23	17-DEC-23
4	keyboard24	13-APR-24
6	keyboard25	13-APR-24

### c. UPDATE

- **FileName + Link:** [UPDATE\\_order.php](#)
- **Line Number Reference for Implementation:** 83, 110, 112
- **SQL command in our code:**

```
$sql = "UPDATE PLACES_ORDER SET ";

$updates = array();
    $updates[] = "USERID = '$new_uid'";
}

if ($td) {
    $new_td = $_POST['insD'];
    $updates[] = "TodayDate = '$new_td'";
}

if ($tc) {
    $new_tc = $_POST['insC'];
    $updates[] = "Total_cost = '$new_tc'";
}

$sql .= implode(", ", $updates);
$sql .= " WHERE OrderID = '$old_oid'";
```

- **Function with the above code:**  
handleUpdateRequest (lines 44 - 120)

## Before Screenshot:

### Update existing order

The values are case sensitive and if you enter in the wrong case, the update statement will not do anything.

To update an order, you must specify the orderId. Blank fields will not be updated and will remain as their initial value.

If you're unsure of which order to update, you can first view the orders by clicking the 'View Orders' button below:

[View Orders](#)

You must fill in at least ONE blank in additional to the OrderID.

OrderId:

UserId:

Date:

Cost:

[Update Order](#)

ORDERID	USERID	TODAYDATE	TOTAL_COST
order1	user1	19-MAR-24	150.99
order2	user2	18-MAR-24	299.99
order3	user3	17-MAR-24	1900.99
order4	user4	16-MAR-24	75.5
order5	user5	15-MAR-24	500
order6	user6	10-FEB-23	650.99
order7	user5	18-AUG-23	82.45
order8	user8	17-DEC-23	365.87
order9	user7	02-FEB-24	64.4
order10	user9	31-JAN-24	200
order11	user10	22-SEP-23	119.62
order12	user2	02-JAN-24	88.88
order13	user8	09-DEC-23	1000.5
order14	user9	01-APR-24	220.05
order15	user3	02-APR-24	103.72
order16	user3	31-JAN-24	200
order17	user3	22-SEP-23	119.62
order18	user3	02-JAN-24	88.88
order19	user8	09-DEC-23	1000.5
order20	user8	01-APR-24	220.05
order21	user8	02-APR-24	103.72
order22	user8	31-JAN-24	200
order23	user7	22-SEP-23	119.62
order24	user7	02-JAN-24	88.88
order25	user7	09-DEC-23	5000.5

## During Screenshot:

## Update existing order

The values are case sensitive and if you enter in the wrong case, the update statement will not do anything.

**To update an order, you must specify the orderId. Blank fields will not be updated and will remain as their initial value.**

If you're unsure of which order to update, you can first view the orders by clicking the 'View Orders' button below:

[View Orders](#)

You must fill in at least ONE blank in additional to the OrderID.

OrderId:

UserId:

Date:

Cost:

[Update Order](#)

**After Screenshot:**

Order updated successfully. ...

ORDERID	USERID	TODAYDATE	TOTAL_COST
order1	user5	01-JAN-24	999
order2	user2	18-MAR-24	299.99
order3	user3	17-MAR-24	1900.99
order4	user4	16-MAR-24	75.5
order5	user5	15-MAR-24	500
order6	user6	10-FEB-23	650.99
order7	user5	18-AUG-23	82.45
order8	user8	17-DEC-23	365.87
order9	user7	02-FEB-24	64.4
order10	user9	31-JAN-24	200
order11	user10	22-SEP-23	119.62
order12	user2	02-JAN-24	88.88
order13	user8	09-DEC-23	1000.5
order14	user9	01-APR-24	220.05
order15	user3	02-APR-24	103.72
order16	user3	31-JAN-24	200
order17	user3	22-SEP-23	119.62
order18	user3	02-JAN-24	88.88
order19	user8	09-DEC-23	1000.5
order20	user8	01-APR-24	220.05
order21	user8	02-APR-24	103.72
order22	user8	31-JAN-24	200
order23	user7	22-SEP-23	119.62
order24	user7	02-JAN-24	88.88
order25	user7	09-DEC-23	5000.5

#### d. SELECTION

- FileName + Link: [select\\_user.php](#)
- Line Number Reference for Implementation: 206-223
- SQL command in our code:

```
$query = "SELECT * FROM USER_NORMALIZED WHERE ";

// Add first condition
if ($attribute1 != 'None' && !empty($val1)) {
    $query .= "$attribute1 = '$val1'";
}

// Add second condition with logical operator
if ($attribute2 != 'None' && !empty($val2)) {
    $query .= " $logicalOperator1 $attribute2 = '$val2'";
}

// Add third condition with logical operator
if ($attribute3 != 'None' && !empty($val3)) {
    $query .= " $logicalOperator2 $attribute3 = '$val3'";
}

$result = executePlainSQL($query);
```

- Function with the above code: handleAttributeChange (lines 91 - 115)

#### Before Screenshot:

Shown below is the USER\_NORMALIZED

#### Select User(s)

Find all Users and their relating information by filtering on the given conditions.

=

AND  
 OR

USERID	EMAIL	EXPERIENCE_LEVEL
user1	ambikamod@hotmail.com	Intermediate
user2	leo@gmail.com	Advanced
user3	ellie@gmail.com	Beginner
user4	jon@ubc.com	Intermediate
user5	essiedavis@gmail.com	Advanced
user6	minag@yahoo.com	Beginner
user7	anniewang@gmail.com	Beginner
user8	jessiej@gmail.com	Advanced
user9	lawrenceholstaff@ubc.com	Intermediate
user10	donolo@gmail.com	Beginner

**During Screenshot:**

### Select User(s)

User ID  =

OR

Experience Level  =

AND  =

**After Screenshot:**

We can see in the table below that we accurately selected for USERID = 'user1' OR EXPERIENCELEVEL = 'Advanced'

User ID	Email Address	Experience Level
user1	ambikamod@hotmail.com	Intermediate
user2	leo@gmail.com	Advanced
user5	essiedavis@gmail.com	Advanced
user8	jessiej@gmail.com	Advanced

e. PROJECTION

- FileName + Link: [keyboard\\_view.php](#)
- Line Number Reference for Implementation: 42
- SQL command in our code:

```
SELECT table_name FROM all_tables WHERE owner = 'ORA_WENDYTSO'
```

- Function with the above code: keyboard\_view.php (lines 40 - 100)

### Before & During Screenshot:

#### Display Selected Attributes from Selected Table

Choose a table: KEYCAP

ACCESORY\_INCLUDED\_ON  
BOARD\_HAS\_LAYOUT  
COLOR\_DETERMINES\_BRIGHTNESS  
COMPLETE\_EXPERIENCE\_DETERMINES\_SALARY  
CONFIGURES  
EMAIL\_DETERMINES\_NAME  
EXPERIENCE\_DETERMINES\_SALARY  
GAMING  
KEYBOARD\_ASSEMBLER\_NORMALIZED  
KEYBOARD\_CONTAINS  
**KEYCAP**  
LIGHTING\_NORMALIZED  
MADE\_BY  
MECHANICAL  
PLACES\_ORDER  
RATING\_HAS\_FEEDBACK  
SWITCHES\_ATTACH\_TO  
USER\_NORMALIZED

#### Display Selected Attributes from Selected Table

Choose attribute(s) for KEYCAP:

- KEYID  
 PRICE  
 MATERIAL  
 BRAND  
 SWITCHID

### After Screenshot:

KEYID	PRICE	MATERIAL	BRAND	SWITCHID
keycap1	2.99	ABS	Generic	1
keycap2	3.99	PBT	Corsair	2
keycap3	4.99	ABS	Razer	3
keycap4	5.99	PBT	SteelSeries	4
keycap5	6.99	ABS	Ducky	5
keycap6	2.99	ABS	Generic	6
keycap7	3.99	PBT	Generic	7
keycap8	4.99	ABS	Razer	8
keycap9	5.99	PBT	Ducky	9
keycap10	6.99	ABS	Ducky	10
keycap11	2.99	ABS	Generic	11
keycap12	3.99	PBT	Ducky	12
keycap13	4.99	ABS	Razer	13
keycap14	5.99	PBT	Generic	14
keycap15	6.99	ABS	Ducky	15

#### f. JOIN

- FileName + Link: [JOIN\\_assemblerRating.php](#)
- Line Number Reference for Implementation: 60-63

- **SQL command:**

```
SELECT k.KeyboardName, r.Rate, bAssemblerID, an.Name, an.Experience_level
FROM RATING_HAS_FEEDBACK r, MADE_BY b, KEYBOARD_CONTAINS k, KEYBOARD_ASSEMBLER_NORMALIZED an
WHERE r.KeyboardName = k.KeyboardName AND r.KeyboardName = b.KeyboardName AND k.KeyboardName = b.KeyboardName AND an.AssemblerId = b.AssemblerId
AND r.Rate = $query";
```

- **Function with the above code:** handleJoinSearchRequest (lines 33 - 68)

**Before Screenshot:**

Below are the RATING\_HAS\_FEEDBACK, MADE\_BY, KEYBOARD\_CONTAINS, and KEYBOARD\_ASSEMBLER\_NORMALIZED which are all used and required to JOIN and answer this query.

RATINGID	RATE	KEYBOARDNAME	TODAYDATE
rating1	9	keyboard1	19-MAR-24
rating2	8	keyboard2	18-MAR-24
rating3	1	keyboard3	17-MAR-24
rating4	2	keyboard4	16-MAR-24
rating5	3	keyboard5	15-MAR-24
rating6	8	keyboard6	10-FEB-23
rating7	9	keyboard7	18-AUG-23
rating8	10	keyboard8	17-DEC-23
rating9	4	keyboard9	02-FEB-24
rating10	5	keyboard10	31-JAN-24
rating11	6	keyboard11	22-SEP-23
rating12	7	keyboard12	02-JAN-24
rating13	8	keyboard13	09-DEC-23
rating14	3	keyboard14	01-APR-24
rating15	2	keyboard15	02-APR-24
rating16	2	keyboard16	10-FEB-23
rating17	4	keyboard17	18-AUG-23
rating18	9	keyboard18	17-DEC-23
rating19	7	keyboard19	02-FEB-24
rating20	6	keyboard20	31-JAN-24
rating21	5	keyboard21	22-SEP-23
rating22	5	keyboard22	02-JAN-24
rating23	8	keyboard23	09-DEC-23
rating24	8	keyboard24	01-APR-24
rating25	4	keyboard25	02-APR-24

ASSEMBLERID	KEYBOARDNAME	DATE_OF_COMPLETION
1	keyboard1	19-MAR-24
2	keyboard2	18-MAR-24
3	keyboard3	17-MAR-24
4	keyboard4	20-MAR-24
5	keyboard5	25-MAR-24
1	keyboard6	26-FEB-24
2	keyboard7	10-SEP-23
3	keyboard8	23-DEC-23
4	keyboard9	19-FEB-24
5	keyboard10	07-MAR-24
1	keyboard11	11-OCT-23
2	keyboard12	17-JAN-24
3	keyboard13	17-DEC-23
4	keyboard14	13-APR-24
5	keyboard15	13-APR-24
1	keyboard16	26-FEB-24
2	keyboard17	10-SEP-23
3	keyboard18	23-DEC-23
4	keyboard19	19-FEB-24
5	keyboard20	07-MAR-24
9	keyboard21	11-OCT-23
2	keyboard22	17-JAN-24
7	keyboard23	17-DEC-23
4	keyboard24	13-APR-24
6	keyboard25	13-APR-24

ASSEMBLERID	NAME	EXPERIENCE_LEVEL	SALARY	NUMBER_OF_COMPLETED_KEYBOARDS
1	Thelonius Ellison	Beginner	800	10
2	Sinatra Golden	Intermediate	2400	20
3	Carl Brunt	Advanced	3200	30
4	Aileen Hoover	Intermediate	2500	25
5	Coraline Alexander	Advanced	3600	35
6	George Mcyntire	Beginner	1800	8
7	Vivian Lee	Intermediate	3400	24
8	Carl Brunt	Advanced	4000	50
9	Aileen Hoover	Intermediate	3100	30
10	Roger Ted	Advanced	3650	30

KEYBOARDNAME	ORDERID	KEYBOARDCONNECTION	TODAYDATE
keyboard1	order1	USB	19-MAR-24
keyboard2	order2	Wireless	18-MAR-24
keyboard3	order3	Bluetooth	17-MAR-24
keyboard4	order4	USB	16-MAR-24
keyboard5	order5	Wireless	15-MAR-24
keyboard6	order6	USB	10-FEB-23
keyboard7	order7	Wireless	18-AUG-23
keyboard8	order8	Bluetooth	17-DEC-23
keyboard9	order9	USB	02-FEB-24
keyboard10	order10	Wireless	31-JAN-24
keyboard11	order11	USB	22-SEP-23
keyboard12	order12	Bluetooth	02-JAN-24
keyboard13	order13	Bluetooth	09-DEC-23
keyboard14	order14	Wireless	01-APR-24
keyboard15	order15	Wireless	02-APR-24
keyboard16	order16	USB	10-FEB-23
keyboard17	order17	Wireless	18-AUG-23
keyboard18	order18	Bluetooth	17-DEC-23
keyboard19	order19	USB	02-FEB-24
keyboard20	order20	Wireless	31-JAN-24
keyboard21	order21	USB	22-SEP-23
keyboard22	order22	Bluetooth	02-JAN-24
keyboard23	order23	Bluetooth	09-DEC-23
keyboard24	order24	Wireless	01-APR-24
keyboard25	order25	Wireless	02-APR-24

#### During Screenshot:

### Find Keyboard Assembler by Rating

Finds the names of the Keyboard Assemblers whom have made keyboards with input Ratings

**Please note that keyboard ratings are on a scale of 1-10.**

Rating:

[View All Tables](#) [Go Back to Main Page](#)

#### After Screenshot:

KEYBOARDNAME	RATE	ASSEMBLERID	NAME	EXPERIENCE_LEVEL
keyboard1	9	1	Thelonius Ellison	Beginner
keyboard7	9	2	Sinatra Golden	Intermediate
keyboard18	9	3	Carl Brunt	Advanced

---

#### g. AGGREGATION WITH GROUP BY

- **FileName + Link:** [AggGB\\_keycapCountByBrand.php](#)
- **Line Number Reference for Implementation:** 34
- **SQL command:**

```
SELECT Brand, Count(*) FROM KEYCAP GROUP BY Brand'
```

- **Function with the above code:** handleAgggbKeycapsReq (lines 31 - 37)

#### Before Screenshot:

Shown below is the KEYCAP table.

KEYID	PRICE	MATERIAL	BRAND	SWITCHID
keycap1	2.99	ABS	Generic	1
keycap2	3.99	PBT	Corsair	2
keycap3	4.99	ABS	Razer	3
keycap4	5.99	PBT	SteelSeries	4
keycap5	6.99	ABS	Ducky	5
keycap6	2.99	ABS	Generic	6
keycap7	3.99	PBT	Generic	7
keycap8	4.99	ABS	Razer	8
keycap9	5.99	PBT	Ducky	9
keycap10	6.99	ABS	Ducky	10
keycap11	2.99	ABS	Generic	11
keycap12	3.99	PBT	Ducky	12
keycap13	4.99	ABS	Razer	13
keycap14	5.99	PBT	Generic	14
keycap15	6.99	ABS	Ducky	15

#### During & After Screenshot:

## Number of Keycaps Grouped By Production Brands

Counts the numbers of keyboards that have use a specific keycap production brand.

[Find keycap brand counts](#)

[View All Tables](#) [Go Back to Main Page](#)

BRAND	COUNT(*)
-------	----------

SteelSeries 1

Razer 3

Ducky 5

Generic 5

Corsair 1

### h. AGGREGATION WITH HAVING

- **FileName + Link:** [AggH\\_assemblerSalaryByEXP.php](#)
- **Line Number Reference for Implementation:** 34-36
- **SQL command:**

```
SELECT Experience_level, MAX(Salary), MIN(Salary), AVG(Salary), Count(*)
  FROM KEYBOARD_ASSEMBLER_NORMALIZED GROUP BY Experience_level
 HAVING MIN(Number of completed keyboards) >= 1";
```

- **Function with the above code:** handleAGGHAssemblerREQ (lines 31 - 39)

**Before Screenshot:**

Shown below is the KEYBOARD\_ASSEMBLER\_NORMALIZED table.

ASSEMBLERID	NAME	EXPERIENCE_LEVEL	SALARY	NUMBER_OF_COMPLETED_KEYBOARDS
1	Thelonius Ellison	Beginner	800	10
2	Sinatra Golden	Intermediate	2400	20
3	Carl Brunt	Advanced	3200	30
4	Aileen Hoover	Intermediate	2500	25
5	Coraline Alexander	Advanced	3600	35
6	George Mcyntire	Beginner	1800	8
7	Vivian Lee	Intermediate	3400	24
8	Carl Brunt	Advanced	4000	50
9	Aileen Hoover	Intermediate	3100	30
10	Roger Ted	Advanced	3650	30

**During & After Screenshot:**

## Salary Summary of Keyboard Assemblers by Experience Level

Determine the max, min, and average salaries of keyboard assemblers as well as the number of them, by their experience level.

[View Salary Summary](#)

[View All Tables](#) [Go Back to Main Page](#)

EXPERIENCE_LEVEL	MAX(SALARY)	MIN(SALARY)	AVG(SALARY)	COUNT(*)
Intermediate	3400	2400	2850	4
Beginner	1800	800	1300	2
Advanced	4000	3200	3612.5	4

### i. NESTED AGGREGATION WITH GROUP BY

- **FileName + Link:** [NESTEDagg\\_assembler.php](#)
- **Line Number Reference for Implementation:** 35-51
- **SQL command:**

```
$sql = "CREATE VIEW sumCost(assembler, sc) AS
    SELECT bAssemblerID, SUM(o.Total_cost)
    FROM MADE_BY b, KEYBOARD_CONTAINS k, KEYBOARD_ASSEMBLER_NORMALIZED an, PLACES_ORDER o
    WHERE o.OrderID = k.OrderID AND k.KeyboardName = b.KeyboardName AND an.AssemblerId = b.AssemblerId
    GROUP BY bAssemblerID";

$result1 = executePlainSQL($sql);

$sql2 = "SELECT s.assembler, s.sc
    FROM sumCost s
    WHERE s.sc < (SELECT AVG(ka.Salary) FROM KEYBOARD_ASSEMBLER_NORMALIZED ka)";

$result2 = executePlainSQL($sql2);

$sql3 = "DROP VIEW sumCost";
executePlainSQL($sql3);
```

- **Function with the above code:** handleNestedAGB (lines 31 - 54)

**Before Screenshot:**

Shown below are the MADE\_BY, KEYBOARD\_CONTAINS, KEYBOARD\_ASSEMBLER\_NORMALIZED, and PLACES\_ORDER table.

ASSEMBLERID	KEYBOARDNAME	DATE_OF_COMPLETION
1	keyboard1	19-MAR-24
2	keyboard2	18-MAR-24
3	keyboard3	17-MAR-24
4	keyboard4	20-MAR-24
5	keyboard5	25-MAR-24
1	keyboard6	26-FEB-24
2	keyboard7	10-SEP-23
3	keyboard8	23-DEC-23
4	keyboard9	19-FEB-24
5	keyboard10	07-MAR-24
1	keyboard11	11-OCT-23
2	keyboard12	17-JAN-24
3	keyboard13	17-DEC-23
4	keyboard14	13-APR-24
5	keyboard15	13-APR-24
1	keyboard16	26-FEB-24
2	keyboard17	10-SEP-23
3	keyboard18	23-DEC-23
4	keyboard19	19-FEB-24
5	keyboard20	07-MAR-24
9	keyboard21	11-OCT-23
2	keyboard22	17-JAN-24
7	keyboard23	17-DEC-23
4	keyboard24	13-APR-24
6	keyboard25	13-APR-24

KEYBOARDNAME	ORDERID	KEYBOARDCONNECTION	TODAYDATE
keyboard1	order1	USB	19-MAR-24
keyboard2	order2	Wireless	18-MAR-24
keyboard3	order3	Bluetooth	17-MAR-24
keyboard4	order4	USB	16-MAR-24
keyboard5	order5	Wireless	15-MAR-24
keyboard6	order6	USB	10-FEB-23
keyboard7	order7	Wireless	18-AUG-23
keyboard8	order8	Bluetooth	17-DEC-23
keyboard9	order9	USB	02-FEB-24
keyboard10	order10	Wireless	31-JAN-24
keyboard11	order11	USB	22-SEP-23
keyboard12	order12	Bluetooth	02-JAN-24
keyboard13	order13	Bluetooth	09-DEC-23
keyboard14	order14	Wireless	01-APR-24
keyboard15	order15	Wireless	02-APR-24
keyboard16	order16	USB	10-FEB-23
keyboard17	order17	Wireless	18-AUG-23
keyboard18	order18	Bluetooth	17-DEC-23
keyboard19	order19	USB	02-FEB-24
keyboard20	order20	Wireless	31-JAN-24
keyboard21	order21	USB	22-SEP-23
keyboard22	order22	Bluetooth	02-JAN-24
keyboard23	order23	Bluetooth	09-DEC-23
keyboard24	order24	Wireless	01-APR-24
keyboard25	order25	Wireless	02-APR-24

ASSEMBLERID	NAME	EXPERIENCE_LEVEL	SALARY	NUMBER_OF_COMPLETED_KEYBOARDS
1	Thelonius Ellison	Beginner	800	10
2	Sinatra Golden	Intermediate	2400	20
3	Carl Brunt	Advanced	3200	30
4	Aileen Hoover	Intermediate	2500	25
5	Coraline Alexander	Advanced	3600	35
6	George Mcyntire	Beginner	1800	8
7	Vivian Lee	Intermediate	3400	24
8	Carl Brunt	Advanced	4000	50
9	Aileen Hoover	Intermediate	3100	30
10	Roger Ted	Advanced	3650	30

ORDERID	USERID	TODAYDATE	TOTAL_COST
order1	user1	19-MAR-24	150.99
order2	user2	18-MAR-24	299.99
order3	user3	17-MAR-24	1900.99
order4	user4	16-MAR-24	75.5
order5	user5	15-MAR-24	500
order6	user6	10-FEB-23	650.99
order7	user5	18-AUG-23	82.45
order8	user8	17-DEC-23	365.87
order9	user7	02-FEB-24	64.4
order10	user9	31-JAN-24	200
order11	user10	22-SEP-23	119.62
order12	user2	02-JAN-24	88.88
order13	user8	09-DEC-23	1000.5
order14	user9	01-APR-24	220.05
order15	user3	02-APR-24	103.72
order16	user3	31-JAN-24	200
order17	user3	22-SEP-23	119.62
order18	user3	02-JAN-24	88.88
order19	user8	09-DEC-23	1000.5
order20	user8	01-APR-24	220.05
order21	user8	02-APR-24	103.72
order22	user8	31-JAN-24	200
order23	user7	22-SEP-23	119.62
order24	user7	02-JAN-24	88.88
order25	user7	09-DEC-23	5000.5

**During & After Screenshot:**

### Keyboard Assembler Salary Discrepancy

Find Keyboard assemblers whose total price of keyboards they have assembled is less than the average salary of all assemblers.

ASSEMBLER	SC
1	1121.6
7	119.62
2	790.94
4	1449.33
5	1023.77
9	103.72

**j. DIVISION**

- FileName + Link: [DIV\\_lightingEffects.php](#)

- Line Number Reference for Implementation: 47-58
- SQL command:

```
SELECT DISTINCT UserID
  FROM PLACES_ORDER o
 WHERE NOT EXISTS
   ((SELECT LightingID
      FROM LIGHTING_NORMALIZED l
     WHERE Effect = 'Wave')
    MINUS
   (SELECT LightingID
      FROM PLACES_ORDER p, CONFIGURES c, KEYBOARD_CONTAINS kc
     WHERE kc.KeyboardName = c.KeyboardName AND
           p.UserID = o.UserID AND
           p.OrderID = kc.OrderID))");
```

- Function with the above code: handleDIV\_waveKeyboards (lines 33 - 61)

**Before Screenshot:**

Shown below are the LIGHTING\_NORMALIZED and CONFIGURES table.

LIGHTINGID	KEYBOARDNAME
1	keyboard1
1	keyboard10
1	keyboard11
1	keyboard15
1	keyboard19
2	keyboard12
2	keyboard16
2	keyboard2
2	keyboard20
3	keyboard13
3	keyboard17
3	keyboard21
3	keyboard3
4	keyboard14
4	keyboard18
4	keyboard22
4	keyboard4
5	keyboard5
6	keyboard6
7	keyboard23
7	keyboard7
8	keyboard24
8	keyboard8
9	keyboard25
9	keyboard9

LIGHTINGID	PRICE	COLOR	EFFECT
1	49.99	RGB	Wave
2	29.99	White	Wave
3	39.99	Blue	Wave
4	19.99	Purple	Wave
5	59.99	RGB	Rainbow
6	49.99	RGB	Breathing
7	29.99	White	Cycle
8	39.99	Blue	Cycle
9	19.99	Purple	Cycle

**During & After Screenshot:****Users who ordered keyboards tha all have the 'Wave' lighting effects configuration**

Find users whose keyboard orders that used all the "Wave" lighting effect configuration.

[Find Users](#)

---

[View All Tables](#) [Go Back to Main Page](#)

**USERID**

user3

user8

---

**6. README**

Here is a [link to the README.txt](#) as per the assignment rubric. Please note that there is "No extra information".