

```

1  -- -----
2  -- -----TABLE CREATION QUERIES-----
3  -- -----
4
5
6  -- -----
7  -- ensure starting brand new tables ----
8  -- -----
9  SET FOREIGN KEY_CHECKS = 0;
10 DROP TABLE IF EXISTS sk8_riders;
11 DROP TABLE IF EXISTS sk8_brand;
12 DROP TABLE IF EXISTS sk8_wheel_type;
13 DROP TABLE IF EXISTS sk8_truck_type;
14 DROP TABLE IF EXISTS sk8_deck_type;
15 DROP TABLE IF EXISTS sk8_wheel_inv;
16 DROP TABLE IF EXISTS sk8_truck_inv;
17 DROP TABLE IF EXISTS sk8_deck_inv;
18 DROP TABLE IF EXISTS sk8_skateboards;
19 DROP TABLE IF EXISTS sk8_riders_skateboards;
20 SET FOREIGN_KEY_CHECKS = 1;
21
22
23 -- -----
24 -- ---Create sk8 Tables-----
25 -- -----
26
27 -- ---Independent Entities-----
28 CREATE TABLE sk8_riders (
29   id INT NOT NULL AUTO_INCREMENT,
30   rider_name varchar(255) NOT NULL,
31   rider_img_url varchar(255) NOT NULL DEFAULT 'http://web.engr.oregonstate.edu/~swansonb/dataFinal/profileshadow.jpg',
32   PRIMARY KEY (id)
33 )ENGINE=InnoDB;
34
35 CREATE TABLE sk8_brand (
36   id INT NOT NULL AUTO_INCREMENT,
37   brand_name varchar(255) NOT NULL,
38   brand_img_url varchar(255),
39   PRIMARY KEY (id),
40   UNIQUE (brand_name)
41 )ENGINE=InnoDB;
42
43
44 -- ---Equipment Types-----
45 CREATE TABLE sk8_wheel_type(
46   id INT NOT NULL AUTO_INCREMENT,
47   wheel_name varchar(255) NOT NULL,
48   fk_brand_id INT NOT NULL,
49   diameter INT NOT NULL,
50   durometer INT NOT NULL,
51   PRIMARY KEY (id),
52   FOREIGN KEY (fk_brand_id) REFERENCES sk8_brand (id)
53 )ENGINE=InnoDB;
54
55 CREATE TABLE sk8_truck_type(
56   id INT NOT NULL AUTO_INCREMENT,
57   truck_name varchar(255) NOT NULL,
58   fk_brand_id INT NOT NULL,
59   width INT NOT NULL,
60   PRIMARY KEY (id),
61   FOREIGN KEY (fk_brand_id) REFERENCES sk8_brand (id)
62 )ENGINE=InnoDB;
63
64 CREATE TABLE sk8_deck_type(
65   id INT NOT NULL AUTO_INCREMENT,
66   deck_name varchar(255) NOT NULL,
67   fk_brand_id INT NOT NULL,
68   length INT NOT NULL,
69   description varchar(511) NOT NULL,
70   PRIMARY KEY (id),
71   FOREIGN KEY (fk_brand_id) REFERENCES sk8_brand (id)
72 )ENGINE=InnoDB;
73
74 -- ---Equipment Inventory-----
75 -- ---Each entry/row in tables is a real world item---
76 CREATE TABLE sk8_wheel_inv(
77   id INT NOT NULL AUTO_INCREMENT,
78   fk_wheel_id INT NOT NULL,
79   color varchar(127),
80   PRIMARY KEY (id),
81   FOREIGN KEY (fk_wheel_id) REFERENCES sk8_wheel_type (id)
82 )ENGINE=InnoDB;
83
84 CREATE TABLE sk8_truck_inv(
85   id INT NOT NULL AUTO_INCREMENT,
86   fk_truck_id INT NOT NULL,
87   PRIMARY KEY (id),
88   FOREIGN KEY (fk_truck_id) REFERENCES sk8_truck_type (id)

```

```

89 )ENGINE=InnoDB;
90
91 CREATE TABLE sk8_deck_inv(
92   id INT NOT NULL AUTO_INCREMENT,
93   fk_deck_id INT NOT NULL,
94   color varchar(127),
95   PRIMARY KEY (id),
96   FOREIGN KEY (fk_deck_id) REFERENCES sk8_deck_type (id)
97 )ENGINE=InnoDB;
98
99 -- -Relational Tables----
100
101 -- a skateboard is a relation of having exactly
102 --   one deck
103 --   one pair of trucks
104 -- & one set of 4 wheels
105 -- each inventory item can only be in one skateboard, hence the unique constraint
106 CREATE TABLE sk8_skateboards(
107   id INT NOT NULL AUTO_INCREMENT,
108   board_name varchar(255) NOT NULL,
109   board_img_url varchar(255) NOT NULL DEFAULT 'http://web.engr.oregonstate.edu/~swansonb/dataFinal/skateboard_line_art.png',
110   fk_deck_id INT NOT NULL,
111   fk_truck_id INT NOT NULL,
112   fk_wheel_id INT NOT NULL,
113   PRIMARY KEY (id),
114   FOREIGN KEY (fk_deck_id) REFERENCES sk8_deck_inv(id),
115   FOREIGN KEY (fk_truck_id) REFERENCES sk8_truck_inv(id),
116   FOREIGN KEY (fk_wheel_id) REFERENCES sk8_wheel_inv(id),
117   UNIQUE (fk_deck_id),
118   UNIQUE (fk_truck_id),
119   UNIQUE (fk_wheel_id),
120   UNIQUE (board_name)
121 )ENGINE=InnoDB;
122
123 -- Many-to-Many pairs of riders and skateboards
124 CREATE TABLE sk8_riders_skateboards(
125   fk_rider_id INT NOT NULL,
126   fk_skateboard_id INT NOT NULL,
127   PRIMARY KEY (fk_rider_id, fk_skateboard_id),
128   FOREIGN KEY (fk_rider_id) REFERENCES sk8_riders(id) ON DELETE CASCADE,
129   FOREIGN KEY (fk_skateboard_id) REFERENCES sk8_skateboards(id) ON DELETE CASCADE
130 )ENGINE=InnoDB;
131
132
133 -- -----
134 -- --Units-----
135 -- -----
136 -- durometer - A
137 -- diameter - mm
138 -- length - inches
139 -- width - inches
140
141 -- -----
142 -- --Populate Tables-----
143 -- -----
144
145 INSERT INTO
146   sk8_brand (brand_name, brand_img_url)
147 VALUES
148   ('Sector 9', 'https://www.edgeboardshop.com/modules/store/attribute_images/609/21520/2720380_med.png'),
149   ('Penny', 'http://skin.pennyskateboards.com/frontend/penny/default/assets/img/logo-greydkdisc.png'),
150   ('Shark Wheels', 'http://www.sharkwheel.com/wp-content/uploads/2015/05/1-Shark-Wheel-Green-Logo.png'),
151   ('Gullwing', 'https://www.edgeboardshop.com/modules/store/attribute_images/609/24574/2647538_med.jpg'),
152   ('Cult Classic', NULL);
153
154 INSERT INTO
155   sk8_riders (rider_name, rider_img_url)
156 VALUES
157   ('Brandon', 'https://lh3.googleusercontent.com/-zMjLf3TGzf4/AAAAAAAAAI/AAAAAAAAABE/waQ_QZ7E7Fs/s120-c/photo.jpg');
158
159 INSERT INTO
160   sk8_riders (rider_name)
161 VALUES
162   ('Doris');
163
164 INSERT INTO
165   sk8_wheel_type (wheel_name,diameter,durometer,fk_brand_id)
166 VALUES
167   ('California Roll',60,78,3),
168   ('CC Longboard Wheels',70,80,5),
169   ('Penny Wheels', 59,79,2);
170
171 INSERT INTO
172   sk8_wheel_inv (fk_wheel_id,color)
173 VALUES
174   (1,'Blue'),
175   (2,'Blue'),
176   (3,'Red'),(3,'Red');
177

```

```

178 INSERT INTO
179 sk8_truck_type (truck_name, width, fk_brand_id)
180 VALUES
181 ('Mission Truck', 9, 4),
182 ('Penny Trucks', 4, 2),
183 ('Penny Trucks', 3, 2);
184
185 INSERT INTO sk8_truck_inv (fk_truck_id) VALUES (1),(2),(3);
186
187 INSERT INTO
188 sk8_deck_type (deck_name,length, description, fk_brand_id)
189 VALUES
190 ('nickel',27,'Plastic Retro Cruiser', 2),
191 ('penny', 22, 'Plastic Retro Mini', 2);
192
193 INSERT INTO
194 sk8_deck_inv (fk_deck_id, color)
195 VALUES
196 (1,'Blue'),
197 (2,'Red/Polka Dot');
198
199 INSERT INTO
200 sk8_skateboards (board_name,fk_deck_id,fk_truck_id,fk_wheel_id)
201 VALUES
202 ('little blue',1,2,1),
203 ('widowmaker',2,3,3);
204
205 INSERT INTO
206 sk8_riders_skateboards(fk_skateboard_id,fk_rider_id)
207 VALUES
208 (1,1);
209
210
211 -- -----
212 -- -----GENERAL USE QUERIES-----
213 -- -----
214
215 -- -----
216 -- -----SELECT QUERIES -----
217 -- -----
218 -- get information for each part in inventory
219 SELECT D.id, D.color as deckColor, DT.deck_name, DT.length, DT.description,
220 B.brand_name as deck_brand_name, B.brand_img_url as deck_brand_img_url, D.fk_deck_id as fkid FROM sk8_deck_inv D
221 INNER JOIN sk8_deck_type DT on D.fk_deck_id = DT.id
222 INNER JOIN sk8_brand B on DT.fk_brand_id = B.id;
223
224 SELECT T.id, TT.truck_name, TT.width,
225 B.brand_name as truck_brand_name, B.brand_img_url as truck_brand_img_url, T.fk_truck_id as fkid FROM sk8_truck_inv T
226 INNER JOIN sk8_truck_type TT on T.fk_truck_id = TT.id
227 INNER JOIN sk8_brand B on TT.fk_brand_id = B.id;
228
229 SELECT W.id, W.color as wheelColor, WT.wheel_name, WT.diameter, WT.durometer,
230 B.brand_name as wheel_brand_name, B.brand_img_url as wheel_brand_img_url, W.fk_wheel_id as fkid FROM sk8_wheel_inv W
231 INNER JOIN sk8_wheel_type WT on W.fk_wheel_id=WT.id
232 INNER JOIN sk8_brand B on WT.fk_brand_id = B.id;
233
234 -- show only available parts in inventory
235 SELECT D.id, D.color as deckColor, DT.deck_name, DT.length, DT.description,
236 B.brand_name as deck_brand_name, B.brand_img_url as deck_brand_img_url, D.fk_deck_id as fkid FROM sk8_deck_inv D
237 INNER JOIN sk8_deck_type DT on D.fk_deck_id = DT.id
238 INNER JOIN sk8_brand B on DT.fk_brand_id = B.id
239 WHERE D.id NOT IN (SELECT fk_deck_id FROM sk8_skateboards);
240
241 SELECT T.id, TT.truck_name, TT.width,
242 B.brand_name as truck_brand_name, B.brand_img_url as truck_brand_img_url, T.fk_truck_id as fkid FROM sk8_truck_inv T
243 INNER JOIN sk8_truck_type TT on T.fk_truck_id = TT.id
244 INNER JOIN sk8_brand B on TT.fk_brand_id = B.id
245 WHERE T.id NOT IN (SELECT fk_truck_id FROM sk8_skateboards);
246
247 SELECT W.id, W.color as wheelColor, WT.wheel_name, WT.diameter, WT.durometer,
248 B.brand_name as wheel_brand_name, B.brand_img_url as wheel_brand_img_url, W.fk_wheel_id as fkid FROM sk8_wheel_inv W
249 INNER JOIN sk8_wheel_type WT on W.fk_wheel_id=WT.id
250 INNER JOIN sk8_brand B on WT.fk_brand_id = B.id
251 WHERE W.id NOT IN (SELECT fk_wheel_id FROM sk8_skateboards);
252
253 --get riders for display
254 SELECT id, rider_name, rider_img_url FROM sk8_riders
255
256 -- select all parts,details, and brands of a skateboard
257 SELECT SK.id, SK.board_name, SK.board_img_url,
258 DT.deck_name, DT.length, DT.description, DI.color as deckColor,
259 DB.deck_brand_name, DB.deck_brand_img_url,
260 TT.truck_name, TT.width, TB.truck_brand_name, TB.truck_brand_img_url,
261 WT.wheel_name, WT.diameter, WT.durometer, WI.color as wheelColor,
262 WB.wheel_brand_name, WB.wheel_brand_img_url
263 FROM sk8_skateboards SK
264 INNER JOIN sk8_deck_inv DI on SK.fk_deck_id = DI.id
265 INNER JOIN sk8_deck_type DT on DI.fk_deck_id = DT.id
266 INNER JOIN sk8_truck_inv TI on SK.fk_truck_id = TI.id

```

```

267 INNER JOIN sk8_truck_type TT ON TT.fk_truck_id = TT.id
268 INNER JOIN sk8_wheel_inv WI ON SK.fk_wheel_id = WI.id
269 INNER JOIN sk8_wheel_type WT ON WI.fk_wheel_id = WT.id
270 INNER JOIN (SELECT WT.id, B.brand_name AS wheel_brand_name, B.brand_img_url AS wheel_brand_img_url FROM sk8_brand B
271 INNER JOIN sk8_wheel_type WT ON WT.fk_brand_id = B.id) WB ON WT.id = WB.id
272 INNER JOIN (SELECT TT.id, B.brand_name AS truck_brand_name, B.brand_img_url AS truck_brand_img_url FROM sk8_brand B
273 INNER JOIN sk8_truck_type TT ON TT.fk_brand_id = B.id) TB ON TT.id = TB.id
274 INNER JOIN (SELECT DT.id, B.brand_name AS deck_brand_name, B.brand_img_url AS deck_brand_img_url FROM sk8_brand B
275 INNER JOIN sk8_deck_type DT ON DT.fk_brand_id = B.id) DB ON DT.id = DB.id;
276
277 -- link together skateboarders and riders
278 SELECT fk_skateboard_id, fk_rider_id, B.board_name FROM sk8_riders skateboards
279 INNER JOIN sk8_skateboards B ON B.id = fk_skateboard_id;
280
281 SELECT fk_skateboard_id, fk_rider_id, R.rider_name FROM sk8_riders skateboards
282 INNER JOIN sk8_riders R ON R.id = fk_rider_id;
283
284 -- create dropdown menus for all possible riders for all skateboards
285 -- using a left outer join on with the riders/skateboard many-to-many table
286 -- and a cross-product table of all riders and skateboards
287 SELECT POSSIBLES.skid, POSSIBLES.board_name, POSSIBLES.rid, POSSIBLES.rider_name
288 FROM (SELECT B.id AS skid, B.board_name, R.id AS rid, R.rider_name
289 FROM sk8_skateboards B INNER JOIN sk8_riders R) POSSIBLES
290 LEFT OUTER JOIN sk8_riders skateboards RS
291 ON RS.fk_rider_id = POSSIBLES.rid AND RS.fk_skateboard_id = POSSIBLES.skid
292 WHERE RS.fk_rider_id IS null;
293
294 -- for making a drop down menu of brands
295 SELECT id, brand_name FROM sk8_brand;
296
297 -- -----
298 -- -----DELETE QUERIES -----
299 -- -----
300
301 DELETE FROM sk8_riders skateboards WHERE fk_skateboard_id = [] AND fk_rider_id = [];
302 DELETE FROM sk8_skateboards WHERE id=[];
303 DELETE FROM sk8_riders WHERE id=[];
304
305 -- deleting an item FROM inventory
306 DELETE FROM sk8_[deck/truck/wheel]_inv WHERE id=[];
307 -- delete type if there are no more instances
308 DELETE FROM sk8_[deck/truck/wheel]_type WHERE id NOT IN (SELECT fk_[deck/truck/wheel]_id FROM sk8_[deck/truck/wheel]_inv);
309 --inform user of board that part is being used in (violating FK constraint)
310 SELECT id, board_name FROM sk8_skateboards WHERE fk_[deck/truck/wheel]_id=[];
311
312 -- -----
313 -- -----INSERT QUERIES -----
314 -- -----
315
316 -- create a new skateboard and rider relationship ---
317 INSERT INTO sk8_riders skateboards(fk_skateboard_id, fk_rider_id) VALUES ([], []);
318
319 -- add a new brand ---
320 INSERT INTO sk8_brand (brand_name, brand_img_url) VALUES ([], []);
321
322 -- add new inventory, if it is a duplicate item then it will reference a type and specify color
323 -- if the item is a brand new type then they type will be created first
324 -- and then the [last id] in the type table used for the foreign key
325 INSERT INTO sk8_deck_inv (fk_deck_id, color) VALUES ([], []);
326 INSERT INTO sk8_deck_type (deck_name, length, description, fk_brand_id) VALUES ([], [], [], []);
327 INSERT INTO sk8_deck_inv (fk_deck_id, color) VALUES ($mysqli->insert_id, []);
328
329 INSERT INTO sk8_truck_inv (fk_truck_id) VALUES ([]);
330 INSERT INTO sk8_truck_type (truck_name, width, fk_brand_id) VALUES ([], [], []);
331 INSERT INTO sk8_truck_inv (fk_truck_id) VALUES ($mysqli->insert_id);
332
333 INSERT INTO sk8_wheel_inv (fk_wheel_id, color) VALUES ([], []);
334 INSERT INTO sk8_wheel_type (wheel_name, diameter, durometer, fk_brand_id) VALUES ([], [], [], []);
335 INSERT INTO sk8_wheel_inv (fk_wheel_id, color) VALUES ($mysqli->insert_id, []);
336
337 -- when creating rider or skateboard image inserted as an UPDATE
338 -- this is in order to check for empty string
339 INSERT INTO sk8_riders (rider_name) VALUES ([]);
340 UPDATE sk8_riders SET rider_img_url=[] WHERE id=[];
341
342 -- -----
343 -- -----UPDATE QUERIES -----
344 -- -----
345
346 UPDATE sk8_skateboards SET board_img_url = DEFAULT WHERE id = [];
347 UPDATE sk8_skateboards SET board_img_url = [] WHERE id=[];
348 UPDATE sk8_riders SET rider_img_url = DEFAULT WHERE id = [];
349 UPDATE sk8_riders SET rider_img_url = [] WHERE id=[];
350
351 -- -----
352 -- -----Queries involved in building a board, a relationship of having 1: deck, truck, wheel -
353 -- -----
354
355 -- create board relationship

```

```

356 INSERT INTO sk8_skateboards (board_name,fk_deck_id,fk_truck_id,fk_wheel_id) VALUES([],[],[],[])
357 -- check for unique constraint error 1062, each real work inventory item can only be used on one board at a time
358 -- if parts are in use, get IDs and Names of those boards to present to user
359 SELECT DISTINCT B.id, B.board_name FROM sk8_skateboards B
360 WHERE fk_deck_id = [] OR fk_truck_id = [] OR fk_wheel_id = [];
361
362 -- if user decides to scrap those boards and assemble the new one
363 -- get boards that have those parts
364 SELECT DISTINCT B.id FROM sk8_skateboards B
365 WHERE fk_deck_id = [] OR fk_truck_id = [] OR fk_wheel_id = [];
366 -- delete (dissassemble) those boards
367 DELETE FROM sk8_skateboards WHERE id=[];
368
369 -- if img_url is not empty string
370 UPDATE sk8_skateboards SET board_img_url=[] WHERE id=$mysqli->insert_id;

```