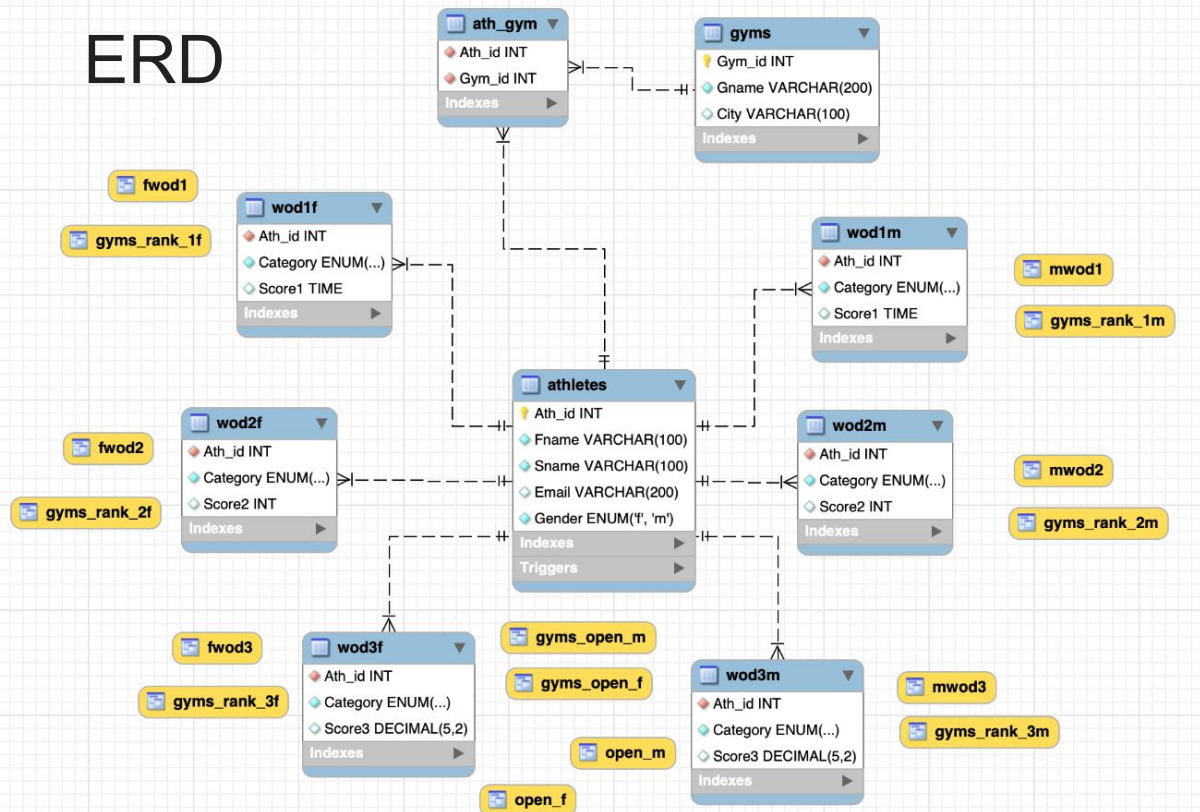


# The CrossFit Open

CFG Data & SQL Project

Alannah Cowley

# ERD



```

CREATE DATABASE crossFit_open;
USE crossfit_0pen;

```

```

CREATE TABLE gyms(
    Gym_id INT NOT NULL PRIMARY KEY,
    Gname VARCHAR(200) NOT NULL,
    City VARCHAR(100));

```

```

CREATE TABLE athletes(
    Ath_id INT NOT NULL PRIMARY KEY,
    Fname VARCHAR(100) NOT NULL,
    Sname VARCHAR(100) NOT NULL,
    Email VARCHAR(200),
    Gender ENUM('f','m') NOT NULL);

```

```

CREATE TABLE ath_gym(
    Ath_id INT NOT NULL,
    Gym_id INT NOT NULL,
    FOREIGN KEY (Ath_id) REFERENCES athletes(Ath_id),
    FOREIGN KEY (Gym_id) REFERENCES gyms(Gym_id));

```

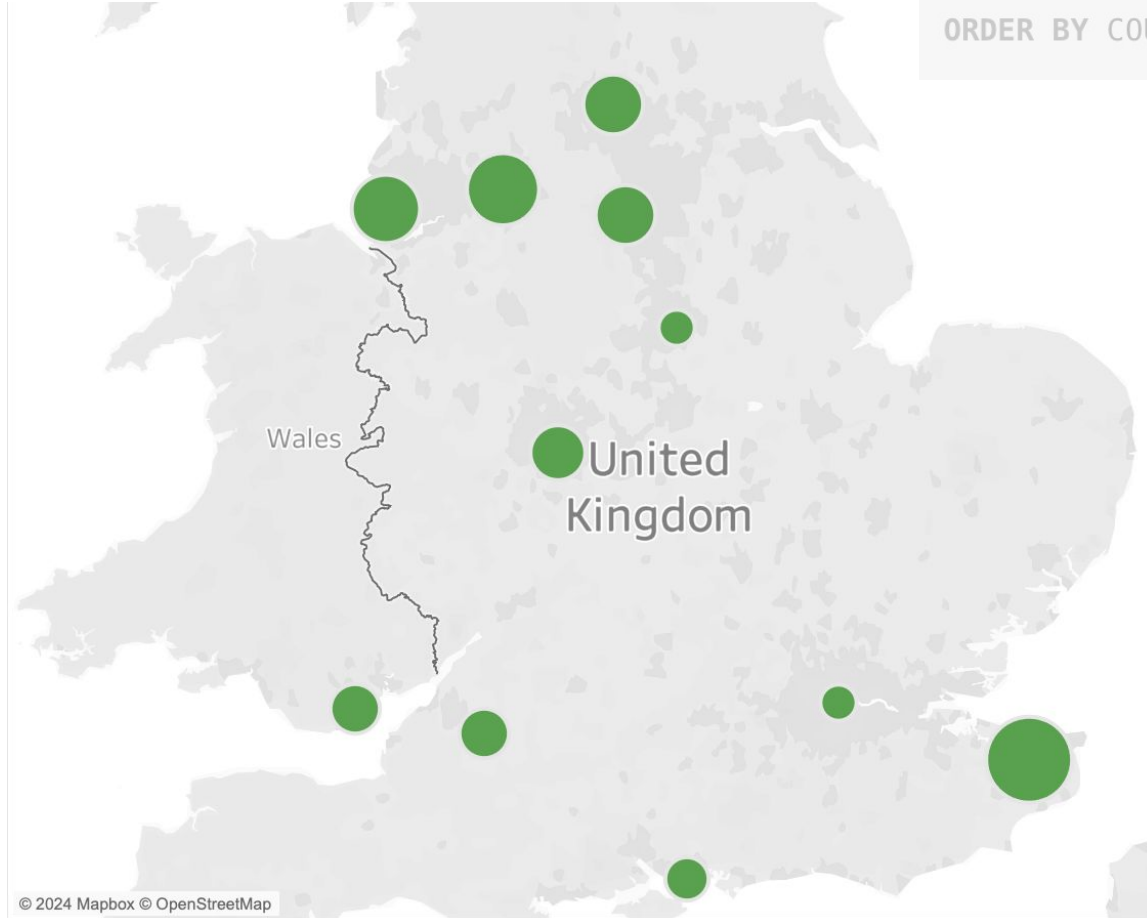
```

CREATE TABLE wod1f(
    Ath_id INT NOT NULL,
    Category ENUM('RX','Ma','Sc') NOT NULL DEFAULT 'Sc',
    Score1 TIME DEFAULT '00:12:01',
    FOREIGN KEY (Ath_id) REFERENCES athletes(Ath_id));

```

# Entries by City

```
SELECT City, COUNT(Ath_id) AS Ath_by_City  
FROM Athlete_Registration  
GROUP BY City  
ORDER BY COUNT(Ath_id) DESC;
```



# Gyms by Entry

```
CREATE VIEW Athlete_Registration AS  
SELECT
```

```
    a.Ath_id,  
    a.FName,  
    a.SName,  
    a.Email,  
    a.Gender,  
    g.GName,  
    g.City,  
    g.Gym_id
```

```
FROM
```

```
Athletes a  
LEFT JOIN Ath_Gym ag  
ON a.Ath_id=ag.Ath_id  
LEFT JOIN Gyms g  
ON ag.Gym_id=g.Gym_id;
```

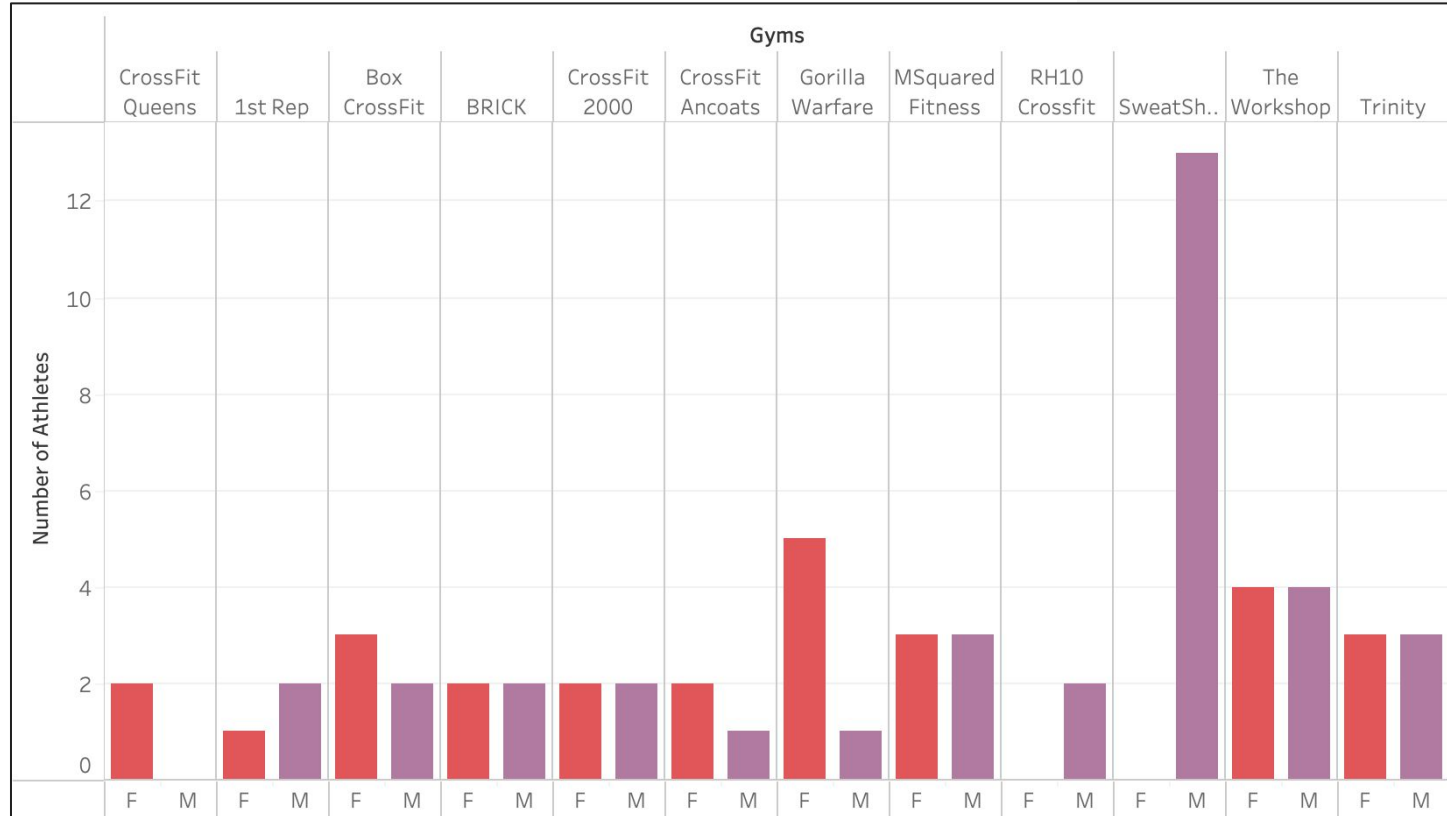
```
SELECT GName, COUNT(Ath_id) AS No_Athletes  
FROM Athlete_Registration  
GROUP BY GName  
HAVING COUNT(Ath_id)>2  
ORDER BY COUNT(Ath_id) DESC;
```



# Male & Female Entries

```

SELECT GName, Gender, COUNT(Ath_id) AS Nof_Athletes
FROM Athlete_Registration
GROUP BY GName, Gender
HAVING Gender='f'
ORDER BY COUNT(Ath_id) DESC;
    
```



# Workout 1

```
CREATE VIEW FWOD1 AS
SELECT
  a.Ath_id,
  a.FName,
  a.SName,
  g.GName,
  g.City,
  1f.Category,
  1f.Score1,
  DENSE_RANK()
    OVER (PARTITION BY 1f.Category ORDER BY 1f.Score1
      AS Rankc1f,
  DENSE_RANK()
    OVER (ORDER BY 1f.Category,1f.Score1 ASC)
      AS Ranka1f
FROM Athletes a
LEFT JOIN WOD1F 1f
ON a.Ath_id=1f.Ath_id
LEFT JOIN Ath_Gym ag
ON a.Ath_id=ag.Ath_id
LEFT JOIN Gyms g
ON ag.Gym_id=g.Gym_id
WHERE a.Gender='f'
ORDER BY Category,Ranka1f;
```

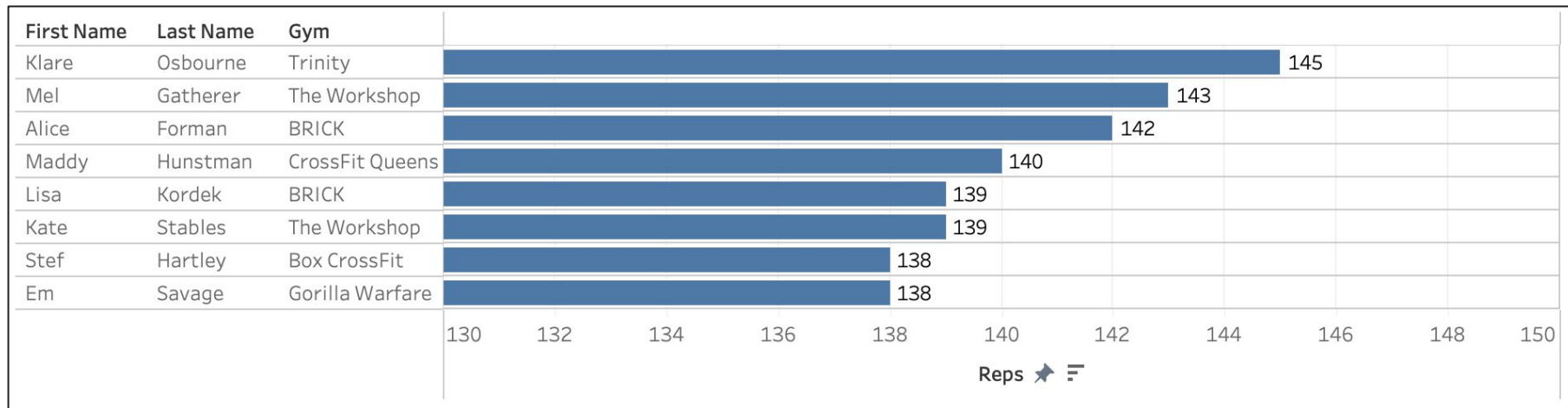
							Score (Time)
Category	Ath id	F Name	S Name	G Name	Rank in Category	Open Rank	
RX	52	Klare	Osbourne	Trinity	1	1	■
	53	Maddy	Hunstman	CrossFit Queens	2	2	■
	43	Em	Savage	Gorilla Warfare	3	3	■
	16	Lisa	Kordek	BRICK	4	4	■
	17	Alice	Forman	BRICK	4	4	■
	21	Mel	Gatherer	The Workshop	5	5	■
	39	Fi	Doin	Gorilla Warfare	6	6	■
	57	Lucy	Truman	CrossFit Queens	7	7	■
	23	Kate	Stables	The Workshop	8	8	■
	42	Emma	Valley	Gorilla Warfare	9	9	■
	10	Stef	Hartley	Box CrossFit	10	10	■
	40	Tess	Doin	Gorilla Warfare	11	11	■
Ma	9	Anna	Penny	Box CrossFit	12	12	■
	58	Lisa	Casper	1st Rep	1	13	■
	49	Laura	Burns	CrossFit Ancoats	2	14	■
	6	Flora	Barton	MSquared Fitness	3	15	■
Sc	19	Ruth	Rivers	The Workshop	4	16	■
	13	Holly	Shanor	Box CrossFit	1	17	■
	50	Laura	Richman	Trinity	2	18	■
	3	Chloe	Brady	MSquared Fitness	3	19	■
	4	Alannah	Cowley	MSquared Fitness	4	20	■
	41	Emma	Fage	Gorilla Warfare	5	21	■
	25	Roisin	Burton	The Workshop	6	22	■
	48	Rita	Soprano	CrossFit Ancoats	7	23	■
	7	Joey	Russell	CrossFit 2000	8	24	■
	56	Moni	Sia	Trinity	9	25	■
	61	Emma	McWheat	CrossFit 2000	10	26	■

# Workout 2

```

SELECT d.Ath_id, d.FName, d.SName, d.GName, d.Score2
FROM FWOD2 d
WHERE d.Score2 >= 138 AND d.Ath_id IN(
    SELECT t.Ath_id FROM(
        SELECT a.Ath_id, a.FName, a.SName, a.GName, a.Category AS CAT
        FROM
        FWOD2 a
        LEFT JOIN FWOD1 b
        ON a.Ath_id=b.Ath_id
        LEFT JOIN FWOD3 c
        ON b.Ath_id=c.Ath_id
        WHERE a.Category=b.Category AND b.Category=c.Category AND a.Category IN
        (SELECT d.Category
        FROM FWOD1 d
        WHERE d.Category='RX')) AS t);

```



# Workout 3

```
//  
CREATE FUNCTION  
    Placed  
    (Rankc INT)  
RETURNS VARCHAR(30)  
DETERMINISTIC  
BEGIN
```

```
    DECLARE Placed VARCHAR(30);  
    IF Rankc>3 THEN  
        SET Placed='No Podium';  
    ELSE IF Rankc=3 THEN  
        SET Placed='Bronze in category';  
    ELSE IF Rankc=2 THEN  
        SET Placed='Silver in category';  
    ELSE IF Rankc=1 THEN  
        SET Placed='Gold in category';  
    END IF;  
    END IF;  
    END IF;  
    END IF;  
    RETURN (Placed);  
END //
```

```
DELIMITER ;
```

Category	Email	F Name	S Name	G Name	Podium	Rankc	Score3
RX	djlance@hotmail.com	Lance	Spinner	1st Rep	Gold in category	1.0	105.0
	jmcgolf@gmail.com	James	McGolfin	BRICK	Silver in category	2.0	102.5
	coachscouse12@hotmail...	James	Wilson	Gorilla Warfare	Bronze in category	3.0	100.0
Ma	paulcasper@gmail.com	Paul	Casper	1st Rep	Gold in category	1.0	92.5
	andyr@hotmail.com	Andy	Rivers	The Workshop	Silver in category	2.0	87.3
	psoprano@sky.com	Paolo	Soprano	CrossFit Ancoats	Bronze in category	3.0	83.8
Sc	leed@hotmail.com	Lee	Davis	MSquared Fitness	Gold in category	1.0	90.0
	gingerscanbat@hotmail.c..	Jonny	Bairstow	SweatShed	Silver in category	2.0	82.5
	notthateric@gmail.com	Eric	Apton	BRICK	Silver in category	2.0	82.5
	kianstudies@gmail.com	Kian	Sia	Trinity	Bronze in category	3.0	80.0



# Open

```
CREATE VIEW Open_F AS
SELECT
    f1.Ath_id,
    f1.FName,
    f1.SName,
    f1.GName,
    f1.City,
    f1.Ranka1f,
    f2.Ranka2f,
    f3.Ranka3f,
    DENSE_RANK() OVER (ORDER BY f1.Ranka1f+f2.Ranka2f+f3.Ranka3f) AS Open_Rank
FROM
FWOD1 f1
LEFT JOIN
FWOD2 f2
ON f1.Ath_id=f2.Ath_id
LEFT JOIN
FWOD3 f3
ON f2.Ath_id=f3.Ath_id;
```

Ath id	F Name	S Name	G Name	
21	Mel	Gatherer	The Workshop	1
17	Alice	Forman	BRICK	2
53	Maddy	Hunstman	CrossFit Queens	2
43	Em	Savage	Gorilla Warfare	3
52	Klare	Osbourne	Trinity	3
16	Lisa	Kordek	BRICK	4
39	Fi	Doin	Gorilla Warfare	5
23	Kate	Stables	The Workshop	6
10	Stef	Hartley	Box CrossFit	7
57	Lucy	Truman	CrossFit Queens	8
40	Tess	Doin	Gorilla Warfare	9
58	Lisa	Casper	1st Rep	10
42	Emma	Valley	Gorilla Warfare	11
49	Laura	Burns	CrossFit Ancoats	12
6	Flora	Barton	MSquared Fitness	13
13	Holly	Shanor	Box CrossFit	14
4	Alannah	Cowley	MSquared Fitness	15
19	Ruth	Rivers	The Workshop	15
9	Anna	Penny	Box CrossFit	16
48	Rita	Soprano	CrossFit Ancoats	17
41	Emma	Fage	Gorilla Warfare	18
3	Chloe	Brady	MSquared Fitness	19
25	Roisin	Burton	The Workshop	20
50	Laura	Richman	Trinity	21
7	Joey	Russell	CrossFit 2000	22
56	Moni	Sia	Trinity	23
61	Emma	McWheat	CrossFit 2000	24

# Open

```
CREATE VIEW Open_F AS
SELECT
```

```
    f1.Ath_id,
    f1.FName,
    f1.SName,
    f1.GName,
    f1.City,
    f1.Ranka1f,
    f2.Ranka2f,
    f3.Ranka3f,
    DENSE_RANK() OVER (ORDER BY f1
```

```
FROM
```

```
FWOD1 f1
```

```
LEFT JOIN
```

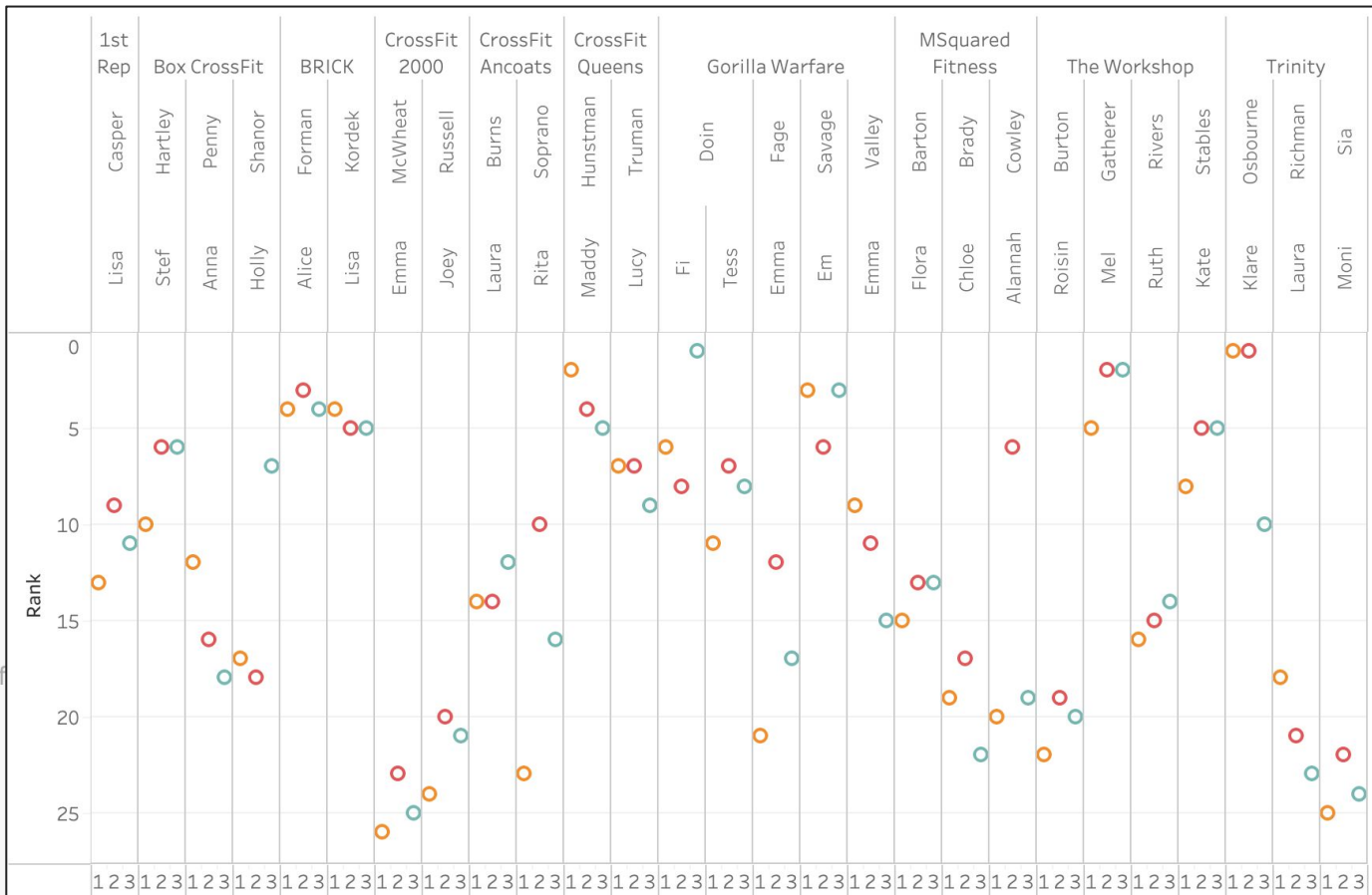
```
FWOD2 f2
```

```
ON f1.Ath_id=f2.Ath_id
```

```
LEFT JOIN
```

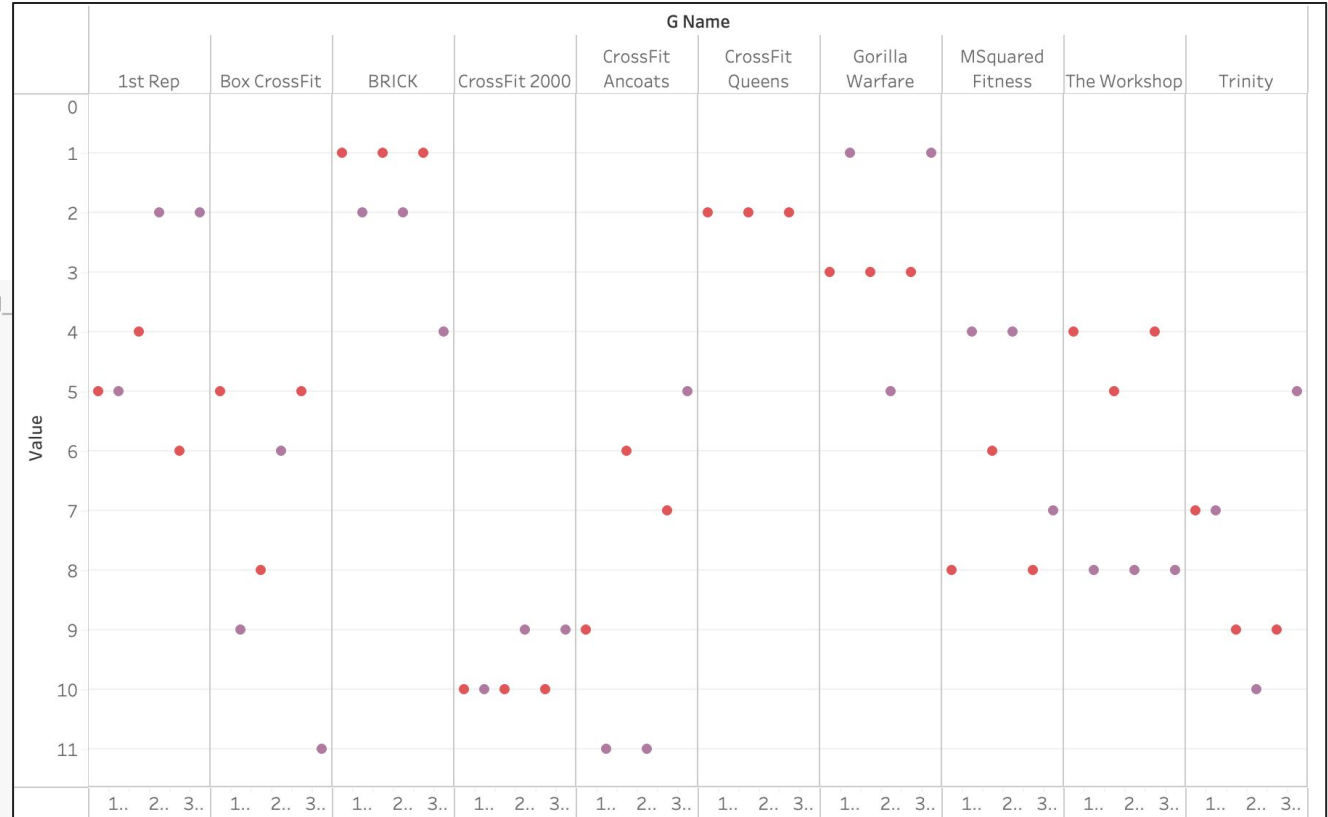
```
FWOD3 f3
```

```
ON f2.Ath_id=f3.Ath_id;
```



# Gyms by Workout

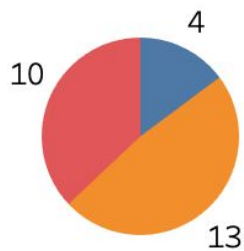
```
SELECT a.GName, a.Gyms_1F_Rank, f.Gyms_1M_
FROM
Gyms_Rank_1f a
LEFT JOIN
Gyms_Rank_2f b
ON a.GName=b.GName
LEFT JOIN
Gyms_Rank_3f c
ON b.GName=c.GName
LEFT JOIN
Gyms_Rank_3m d
ON d.GName=c.GName
LEFT JOIN
Gyms_Rank_2m e
ON d.GName=e.GName
LEFT JOIN
Gyms_Rank_1m f
ON f.GName=e.GName;
```



# Categories by Workout



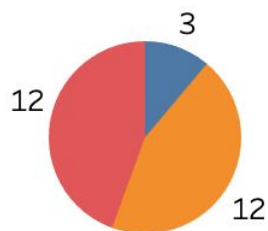
1F



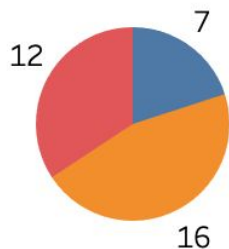
2F



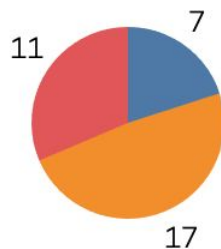
3F



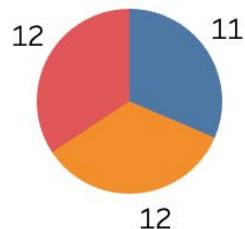
1M



2M



3M



- `SET sql_mode = 'TRADITIONAL';`
- `SELECT Category, COUNT(*) AS CatCount  
FROM WOD1F  
GROUP BY Category;`
- `SELECT Category, COUNT(*) AS CatCount  
FROM WOD2F  
GROUP BY Category;`
- `SELECT Category, COUNT(*) AS CatCount  
FROM WOD3F  
GROUP BY Category;`
- `SELECT Category, COUNT(*) AS CatCount  
FROM WOD1M  
GROUP BY Category;`
- `SELECT Category, COUNT(*) AS CatCount  
FROM WOD2M  
GROUP BY Category;`
- `SELECT Category, COUNT(*) AS CatCount  
FROM WOD3M  
GROUP BY Category;`

# Scores to Beat & Reach New Athlete Triggers

```
CREATE PROCEDURE Score_To_Beat
(
  B
  FROM
  WOD1F
  WHERE WOD1F.Category = in_category;
  ELSE IF in_WOD='1M'
  THEN
  SELECT
    MIN(Score1) AS Score_To_Beat
  FROM
```

```
CALL Score_To_Beat('2M', 'Sc'); /* His score is 159 reps */
```

```
CALL Score_To_Reach('2M'); /* The least score in RX is 119 reps, so he should definitely give it a go at RX weight.*/
```

```
in_WOD='2F'
THEN
SELECT
  MAX(Score2) AS Score_To_Reach
FROM
WOD2F
WHERE WOD2F.Category = in_category;
ELSE IF
in_WOD='2M'
THEN
SELECT
  MAX(Score2) AS Score_To_Reach
FROM
WOD2M
WHERE WOD2M.Category = in_category;
```

```
CALL New_Athlete (61, 'Emma', 'McWheat', 'emmamcw@hotmail.com', 'f');
CALL New_Athlete (62, 'Ian', 'McWheat', 'ianmcw@hotmail.com', 'm');

INSERT INTO Ath_Gym
(Ath_id, Gym_id)
VALUES
(61, 7),
(62, 7);
```

```
DELIMITER //
```

```
CREATE TRIGGER New_Ath1
AFTER INSERT
ON Athletes
FOR EACH ROW
BEGIN
  IF NEW.Gender = 'f' THEN
    INSERT INTO WOD1F (Ath_id) VALUES (NEW.Ath_id);
  ELSEIF NEW.Gender = 'm' THEN
    INSERT INTO WOD1M (Ath_id) VALUES (NEW.Ath_id);
  END IF;
END;
//
```

```
ON Athletes
FOR EACH ROW
BEGIN
  IF NEW.Gender = 'f' THEN
    INSERT INTO WOD2F (Ath_id) VALUES (NEW.Ath_id);
  ELSEIF NEW.Gender = 'm' THEN
    INSERT INTO WOD2M (Ath_id) VALUES (NEW.Ath_id);
  END IF;
END;
//
```