**Table of Contents**

[Query 1: aggregate highest rank](#pg4skbqs9sx1)

[Query 2: division skins not bought](#f2lx5kexasmd)

[Query 3: delete account](#9tacv2ek65b5)

[Query 4: update reset rank](#l40ld1atjkvp)

[Query 5: join skins and owns](#9dcxtfw8usua)

[Query 6: aggregate most expensive skin](#smbs2tz4mpco)

[Query 7: aggregate most spending](#ewwm34rsbmia)

[Query 8: division owns all heroes already](#uju7e618wej1)

[Query 9: nested aggregate](#svcuuysh7m0l)

[Query 10: nested aggregate](#hitld3nquoso)

[Database Creation DDL](#x6jt9f8in9ma)

[**Find the region with the most high ranked players (aggregate)**](#lskmzdgn02w0)

DROP VIEW HighRank;

CREATE VIEW HighRank AS

/\* return players and their region \*/

SELECT gameName, seasonRank, region, city

FROM Servers, Accounts

WHERE Servers.id = Accounts.serverId

/\* if player has a high rank \*/

AND seasonRank > 20;

SELECT \* FROM HighRank;

DROP VIEW BestRegion;

CREATE VIEW BestRegion AS

/\* return region and number of high ranked players in each region \*/

SELECT region, count(region) AS goodPlayers

FROM HighRank

GROUP BY region;

SELECT \* FROM BestRegion;

[**Find the skins that a player does not own yet for a particular hero (division)**](#lskmzdgn02w0)

DROP VIEW OwnsSkins;

CREATE VIEW OwnsSkins AS

/\* return useful results of join between Owns and Skins \*/

SELECT gameName, heroName, theme, rarity

FROM Owns

JOIN Skins

ON Owns.productId = Skins.productId

ORDER BY gameName;

SELECT \* FROM OwnsSkins;

DROP VIEW ManySkins;

CREATE VIEW ManySkins AS

/\* return players who own lots of skins \*/

SELECT gameName, count(gameName) AS numSkins

FROM OwnsSkins

GROUP BY gameName

HAVING count(gameName) >= 4;

SELECT \* FROM ManySkins;

DROP VIEW SkinsNotOwned;

CREATE VIEW SkinsNotOwned AS

SELECT gameName, heroName, theme, rarity

FROM Accounts, Skins

/\* find skins not owned by that player \*/

WHERE Skins.theme NOT IN (

SELECT theme

FROM OwnsSkins

/\* for each gameName \*/

WHERE Accounts.gameName = OwnsSkins.gameName

/\* for each hero \*/

AND Skins.heroName = OwnsSkins.heroName

) /\* players who have many skins \*/

AND Accounts.gameName IN (

SELECT gameName

FROM ManySkins

)

ORDER BY gameName, heroName;

SELECT \* FROM SkinsNotOwned;

[**Delete an account and all of the things it owns and all the records of games it has played (delete)**](#lskmzdgn02w0)

DELETE FROM Accounts

WHERE gameName = 'Pancake Bunny';

SELECT \* FROM Accounts;

SELECT \* FROM Owns;

SELECT \* FROM MatchHistory;

[**Decrease ranks of all players when a new competition starts (update)**](#lskmzdgn02w0)

UPDATE Accounts

/\* decrease all players' ranks by 20% and subtract 4

\* so all players' ranks will go down by a lot since max rank is 27 \*/

SET seasonRank = FLOOR(seasonRank \* 0.80 - 4);

SELECT gameName, seasonRank FROM Accounts;

UPDATE Accounts

/\* if player rank is negative then set it to one \*/

SET seasonRank = 1

WHERE seasonRank <= 0;

SELECT gameName, seasonRank FROM Accounts;

[**Number of skins for each hero (join)**](#lskmzdgn02w0)

DROP VIEW numberSkins;

CREATE VIEW numberSkins AS

/\* return the name of hero, number of skins that hero has \*/

SELECT name AS HeroName, count(name) AS NumberOfSkins

FROM heroes

JOIN skins

ON heroes.name = skins.heroname

/\* group by each hero \*/

GROUP BY name

ORDER BY name;

SELECT \* FROM numberSkins;

[**Most expensive skin for each hero (aggregate)**](#lskmzdgn02w0)

DROP VIEW MaxPrice;

CREATE VIEW MaxPrice AS

/\* return the price of the most expensive skin \*/

SELECT heroName, max(price) AS maxPrice

/\* get table of hero skin information \*/

FROM products

JOIN skins

ON products.Id = skins.productId

/\* group into categories of each hero \*/

GROUP BY heroName;

SELECT \* FROM MaxPrice;

DROP VIEW ProductsSkins;

CREATE VIEW ProductsSkins AS

/\* to store results of join on products and skins tables \*/

SELECT \*

FROM products, skins

WHERE products.Id = skins.productId;

DROP VIEW MaxPriceSkin;

CREATE VIEW MaxPriceSkin AS

/\* return each skin found and the details about each skin \*/

SELECT ProductsSkins.heroName, theme, rarity, price

/\* get table of hero skin info and most expensive skin info \*/

FROM ProductsSkins, MaxPrice

/\* for each hero \*/

WHERE MaxPrice.heroName = ProductsSkins.heroName

/\* find the skins whose price equals the price of the most expensive skin that each hero owns \*/

AND MaxPrice.maxPrice = ProductsSkins.price;

SELECT \* FROM MaxPriceSkin;

[**Person who spends the most (aggregate)**](#lskmzdgn02w0)

DROP VIEW ProductsOwns;

CREATE VIEW ProductsOwns AS

SELECT \*

/\* get table of products that each account owns \*/

FROM products

JOIN owns

ON products.id = owns.productId;

SELECT \* FROM ProductsOwns;

DROP VIEW SpendHistory;

CREATE VIEW SpendHistory AS

/\* get the account name, total spend amount, number of purchases \*/

SELECT gameName, sum(price) AS totalSpending, count(price) AS numPurchases

FROM ProductsOwns

/\* for game items that must be bought with real money \*/

WHERE price > 0

GROUP BY gameName;

SELECT \* FROM SpendHistory;

[**Person who already owns all the heroes (division)**](#lskmzdgn02w0)

DROP VIEW NotOwn;

CREATE VIEW NotOwn AS

SELECT DISTINCT player.gameName, Heroes.name AS heroName

FROM Heroes, Owns player

/\* get heroes player does not own\*/

WHERE Heroes.name NOT IN (

/\* get the heroes player owns \*/

SELECT Heroes.name

FROM Owns

JOIN Heroes

ON Heroes.productId = Owns.productId

/\* for each player \*/

WHERE player.gameName = Owns.gameName

);

SELECT \* FROM NotOwn;

DROP VIEW OwnsAll;

CREATE VIEW OwnsAll AS

/\* return the name of the player \*/

SELECT DISTINCT player.gameName

FROM Owns player

/\* if the list of heroes is empty \*/

WHERE NOT EXISTS (

/\* get all the heroes a player does not own \*/

SELECT heroName

FROM NotOwn

/\* for each player \*/

WHERE player.gameName = NotOwn.gameName

);

SELECT \* FROM OwnsAll;

[**Find the average price of the products, for each Gamename for which their average product price is higher than the average price of all they already have. (nested aggregation: average)**](#lskmzdgn02w0)

DROP VIEW AveragePurchase;

CREATE VIEW ProductsOwns AS

SELECT \*

/\* get table of products that each account owns \*/

FROM products

JOIN owns

ON products.id = owns.productId;

DROP VIEW AveragePurchase;

CREATE VIEW AveragePurchase AS

SELECT gameName, AVG(price) AS averagePrice

FROM ProductsOwns

/\* get players whose average price of all things they bought

\* is higher than the game's average \*/

GROUP BY gameName

HAVING AVG(price) >(

/\* return average price of all products in the entire game \*/

SELECT AVG(price)

FROM ProductsOwns

);

SELECT \* FROM AveragePurchase;

[**Find the price of the cheapest price with price > 900, for each Gamename with at least 7 PRODUCTID (nested aggregation: count and average).**](#lskmzdgn02w0)

DROP VIEW ProductsOwns;

CREATE VIEW ProductsOwns AS

SELECT \*

/\* get table of products that each account owns \*/

FROM products

JOIN owns

ON products.Id = owns.productId;

DROP VIEW PriceFilter;

CREATE VIEW PriceFilter AS

/\* return the average price of items bought for each player \*/

SELECT gameName, MIN(price) AS minPrice

FROM ProductsOwns

/\* get players who have bought more than 7 products \*/

GROUP BY gameName

HAVING 7 < count(price);

SELECT \* FROM PriceFilter;

[**Database Creation DDL**](#lskmzdgn02w0)

https://www.tutorialspoint.com/oracle\_terminal\_online.php

https://learn.uq.edu.au/bbcswebdav/pid-3566639-dt-content-rid-15011891\_1/courses/INFS7901S\_6820\_20147/INFS7901%20project%20.pdf

DROP TABLE MatchHistory CASCADE constraints;

DROP TABLE Skins CASCADE constraints;

DROP TABLE Heroes CASCADE constraints;

DROP TABLE Products CASCADE constraints;

DROP TABLE Owns CASCADE constraints;

DROP TABLE Accounts CASCADE constraints;

DROP TABLE Servers CASCADE constraints;

CREATE TABLE Servers (

id INT NOT NULL,

city VARCHAR(100) NOT NULL,

region VARCHAR(100) NOT NULL,

PRIMARY KEY (id)

);

INSERT INTO Servers VALUES (1, 'Chicago', 'North America');

INSERT INTO Servers VALUES (2, 'New York', 'North America');

INSERT INTO Servers VALUES (3, 'Sydney', 'Oceania');

INSERT INTO Servers VALUES (4, 'Melbourne', 'Oceania');

INSERT INTO Servers VALUES (5, 'Canberra', 'Oceania');

CREATE TABLE Accounts (

gameName VARCHAR(50) NOT NULL,

password VARCHAR(50) NOT NULL,

balanceBlue INT DEFAULT 0,

balanceRiot INT DEFAULT 0,

seasonRank INT,

serverID INT NOT NULL,

PRIMARY KEY (gameName),

FOREIGN KEY (serverID) REFERENCES Servers(id) ON DELETE CASCADE

);

INSERT INTO Accounts VALUES ('Puffy Snowflakes', 'puffy', 4000, 1000, 27, 4);

INSERT INTO Accounts VALUES ('Pancake Bunny', 'pancake', 2000, 3000, 9, 2);

INSERT INTO Accounts VALUES ('Will of the Son', 'will', 1000, 1000, 20, 3);

INSERT INTO Accounts VALUES ('Mattress World', 'mattress', 2000, 2000, 14, 3);

INSERT INTO Accounts VALUES ('Key Key', 'key', 70000, 15000, 1, 3);

INSERT INTO Accounts VALUES ('Angry Butterstick', 'angry', 4000, 1000, 21, 1);

INSERT INTO Accounts VALUES ('Cute Tootsie Roll', 'cute', 4000, 1000, 22, 2);

INSERT INTO Accounts VALUES ('I Bake Little Kids', 'ibake', 4000, 1000, 23, 4);

INSERT INTO Accounts VALUES ('The Dark Avenger', 'thedark', 4000, 1000, 24, 2);

INSERT INTO Accounts VALUES ('Shade of Noxius', 'shade', 4000, 1000, 26, 1);

INSERT INTO Accounts VALUES ('Stale Peanut Thrower', 'stale', 4000, 1000, 21, 2);

INSERT INTO Accounts VALUES ('Mega Pie Eater', 'mega', 4000, 1000, 22, 3);

INSERT INTO Accounts VALUES ('Sharp Popsicle', 'sharp', 4000, 1000, 23, 4);

INSERT INTO Accounts VALUES ('Who Stole My Face', 'who', 4000, 1000, 24, 1);

CREATE TABLE Products (

id INT NOT NULL,

price INT NOT NULL,

PRIMARY KEY (id)

);

INSERT INTO products VALUES (1, 260);

INSERT INTO products VALUES (2, 260);

INSERT INTO products VALUES (3, 880);

INSERT INTO products VALUES (4, 880);

INSERT INTO products VALUES (5, 880);

INSERT INTO products VALUES (6, 975);

INSERT INTO products VALUES (7, 750);

INSERT INTO products VALUES (8, 1820);

INSERT INTO products VALUES (9, 1350);

INSERT INTO products VALUES (10, 1350);

INSERT INTO products VALUES (11, 1350);

INSERT INTO products VALUES (12, 0);

INSERT INTO products VALUES (13, 0);

INSERT INTO products VALUES (14, 0);

INSERT INTO products VALUES (15, 0);

INSERT INTO products VALUES (16, 0);

INSERT INTO products VALUES (17, 0);

INSERT INTO Products VALUES (18, 520);

INSERT INTO Products VALUES (19, 1350);

INSERT INTO Products VALUES (20, 1350);

INSERT INTO Products VALUES (21, 1350);

INSERT INTO Products VALUES (22, 1820);

CREATE TABLE Owns (

gameName VARCHAR(50) NOT NULL,

productID INT NOT NULL,

purchaseDate TIMESTAMP NOT NULL,

PRIMARY KEY (gameName, productID),

FOREIGN KEY (gameName) REFERENCES Accounts(gameName) ON DELETE CASCADE,

FOREIGN KEY (productID) REFERENCES Products(id) ON DELETE CASCADE

);

INSERT INTO Owns VALUES ('Puffy Snowflakes', 1, TIMESTAMP '2018-03-24 18:35:46');

INSERT INTO Owns VALUES ('Puffy Snowflakes', 2, TIMESTAMP '2018-03-24 18:35:46');

INSERT INTO Owns VALUES ('Puffy Snowflakes', 3, TIMESTAMP '2018-03-24 18:35:46');

INSERT INTO Owns VALUES ('Puffy Snowflakes', 4, TIMESTAMP '2018-03-24 18:35:46');

INSERT INTO Owns VALUES ('Puffy Snowflakes', 5, TIMESTAMP '2018-03-24 18:35:46');

INSERT INTO Owns VALUES ('Pancake Bunny', 1, TIMESTAMP '2018-03-21 22:46:26');

INSERT INTO Owns VALUES ('Pancake Bunny', 2, TIMESTAMP '2018-03-21 22:46:26');

INSERT INTO Owns VALUES ('Pancake Bunny', 3, TIMESTAMP '2018-03-21 22:46:26');

INSERT INTO Owns VALUES ('Pancake Bunny', 4, TIMESTAMP '2018-03-21 22:46:26');

INSERT INTO Owns VALUES ('Will of the Son', 1, TIMESTAMP '2018-03-22 19:24:37');

INSERT INTO Owns VALUES ('Will of the Son', 3, TIMESTAMP '2018-03-22 19:24:37');

INSERT INTO Owns VALUES ('Will of the Son', 5, TIMESTAMP '2018-03-22 21:24:37');

INSERT INTO Owns VALUES ('Mattress World', 1, TIMESTAMP '2018-03-20 23:48:32');

INSERT INTO Owns VALUES ('Mattress World', 4, TIMESTAMP '2018-03-20 23:48:32');

INSERT INTO Owns VALUES ('Key Key', 1, TIMESTAMP '2018-03-24 11:27:32');

INSERT INTO Owns VALUES ('Key Key', 2, TIMESTAMP '2018-03-24 11:27:32');

INSERT INTO Owns VALUES ('Key Key', 3, TIMESTAMP '2018-03-24 11:27:32');

INSERT INTO Owns VALUES ('Key Key', 4, TIMESTAMP '2018-03-24 11:27:32');

INSERT INTO Owns VALUES ('Key Key', 5, TIMESTAMP '2018-03-24 11:27:32');

INSERT INTO Owns VALUES ('Key Key', 6, TIMESTAMP '2018-03-24 11:27:32');

INSERT INTO Owns VALUES ('Puffy Snowflakes', 7, TIMESTAMP '2018-03-24 18:35:46');

INSERT INTO Owns VALUES ('Puffy Snowflakes', 8, TIMESTAMP '2018-03-24 18:35:46');

INSERT INTO Owns VALUES ('Puffy Snowflakes', 9, TIMESTAMP '2018-03-24 18:35:46');

INSERT INTO Owns VALUES ('Puffy Snowflakes', 10, TIMESTAMP '2018-03-24 18:35:46');

INSERT INTO Owns VALUES ('Pancake Bunny', 7, TIMESTAMP '2018-03-21 22:46:26');

INSERT INTO Owns VALUES ('Pancake Bunny', 8, TIMESTAMP '2018-03-21 22:46:26');

INSERT INTO Owns VALUES ('Pancake Bunny', 9, TIMESTAMP '2018-03-21 22:46:26');

INSERT INTO Owns VALUES ('Pancake Bunny', 10, TIMESTAMP '2018-03-21 22:46:26');

INSERT INTO Owns VALUES ('Will of the Son', 7, TIMESTAMP '2018-03-22 19:24:37');

INSERT INTO Owns VALUES ('Will of the Son', 9, TIMESTAMP '2018-03-22 19:24:37');

INSERT INTO Owns VALUES ('Will of the Son', 11, TIMESTAMP '2018-03-22 19:24:37');

INSERT INTO Owns VALUES ('Will of the Son', 12, TIMESTAMP '2018-03-22 19:24:37');

INSERT INTO Owns VALUES ('Mattress World', 7, TIMESTAMP '2018-03-20 23:48:32');

INSERT INTO Owns VALUES ('Mattress World', 10, TIMESTAMP '2018-03-20 23:48:32');

INSERT INTO Owns VALUES ('Mattress World', 11, TIMESTAMP '2018-03-20 23:48:32');

INSERT INTO Owns VALUES ('Key Key', 7, TIMESTAMP '2018-03-24 11:27:32');

INSERT INTO Owns VALUES ('Key Key', 8, TIMESTAMP '2018-03-24 11:27:32');

INSERT INTO Owns VALUES ('Key Key', 9, TIMESTAMP '2018-03-24 11:27:32');

INSERT INTO Owns VALUES ('Key Key', 10, TIMESTAMP '2018-03-24 11:27:32');

INSERT INTO Owns VALUES ('Key Key', 11, TIMESTAMP '2018-03-24 11:27:32');

INSERT INTO Owns VALUES ('Key Key', 12, TIMESTAMP '2018-03-24 11:27:32');

CREATE TABLE Heroes (

name VARCHAR(50) NOT NULL ,

class VARCHAR(50) NOT NULL,

productID INT NOT NULL,

PRIMARY KEY (name),

FOREIGN KEY (productID) REFERENCES Products(id) ON DELETE CASCADE

);

INSERT INTO Heroes VALUES ('Amumu', 'Tank', 1);

INSERT INTO Heroes VALUES ('Ashe', 'Marksman', 2);

INSERT INTO Heroes VALUES ('Vladimir', 'Mage', 3);

INSERT INTO Heroes VALUES ('Fizz', 'Assassin', 4);

INSERT INTO Heroes VALUES ('Fiora', 'Skirmisher', 5);

INSERT INTO Heroes VALUES ('Rakan', 'Support', 6);

CREATE TABLE Skins (

theme VARCHAR(50) DEFAULT 'Classic',

heroName VARCHAR(50) NOT NULL,

productID INT NOT NULL,

rarity VARCHAR(20) NOT NULL,

PRIMARY KEY (theme, heroName),

FOREIGN KEY (productID) REFERENCES Products(id) ON DELETE CASCADE,

FOREIGN KEY (heroName) REFERENCES Heroes(name) ON DELETE CASCADE

);

INSERT INTO Skins VALUES ('Little Knight', 'Amumu', 7, 'Superior');

INSERT INTO Skins VALUES ('Project', 'Ashe', 8, 'Legendary');

INSERT INTO Skins VALUES ('Soulstealer', 'Vladimir', 9, 'Epic');

INSERT INTO Skins VALUES ('Super Galaxy', 'Fizz', 10, 'Epic');

INSERT INTO Skins VALUES ('Project', 'Fiora', 11, 'Legendary');

INSERT INTO Skins VALUES ('Classic', 'Amumu', 12, 'Classic');

INSERT INTO Skins VALUES ('Classic', 'Ashe', 13, 'Classic');

INSERT INTO Skins VALUES ('Classic', 'Vladimir', 14, 'Classic');

INSERT INTO Skins VALUES ('Classic', 'Fizz', 15, 'Classic');

INSERT INTO Skins VALUES ('Classic', 'Fiora', 16, 'Classic');

INSERT INTO Skins VALUES ('Classic', 'Rakan', 17, 'Classic');

INSERT INTO Skins VALUES ('Royal Guard', 'Fiora', 18, 'Superior');

INSERT INTO Skins VALUES ('Pool Party', 'Fiora', 19, 'Epic');

INSERT INTO Skins VALUES ('Soaring Sword', 'Fiora', 20, 'Epic');

INSERT INTO Skins VALUES ('Academy', 'Vladimir', 21, 'Epic');

INSERT INTO Skins VALUES ('Blood Lord', 'Vladimir', 22, 'Legendary');

CREATE TABLE MatchHistory (

gameName VARCHAR(50) NOT NULL,

startTimestamp TIMESTAMP NOT NULL,

endTimestamp TIMESTAMP NOT NULL,

result VARCHAR(5) NOT NULL,

partySize INT DEFAULT 1,

position VARCHAR(50) NOT NULL,

heroName VARCHAR(50) NOT NULL,

skinTheme VARCHAR(50) DEFAULT 'Classic',

PRIMARY KEY (gameName, startTimestamp),

FOREIGN KEY (gameName) REFERENCES Accounts(gameName) ON DELETE CASCADE,

FOREIGN KEY (heroName) REFERENCES Heroes(name) ON DELETE CASCADE,

FOREIGN KEY (skinTheme, heroName) REFERENCES Skins(theme, heroName) ON DELETE CASCADE

);

INSERT INTO MatchHistory VALUES ('Will of the Son', TIMESTAMP '2018-03-22 20:00:00', TIMESTAMP '2018-03-22 22:40:00',

'Win', 1, 'Top', 'Vladimir', 'Classic');

INSERT INTO MatchHistory VALUES ('Will of the Son', TIMESTAMP '2018-03-22 21:00:00', TIMESTAMP '2018-03-22 21:40:00',

'Win', 1, 'Top', 'Vladimir', 'Classic');

INSERT INTO MatchHistory VALUES ('Will of the Son', TIMESTAMP '2018-03-22 22:00:00', TIMESTAMP '2018-03-22 22:40:00',

'Win', 1, 'Top', 'Vladimir', 'Soulstealer');

INSERT INTO MatchHistory VALUES ('Will of the Son', TIMESTAMP '2018-03-23 12:00:00', TIMESTAMP '2018-03-22 12:40:00',

'Loss', 1, 'Marksman', 'Ashe', 'Project');

INSERT INTO MatchHistory VALUES ('Mattress World', TIMESTAMP '2018-03-23 13:00:00', TIMESTAMP '2018-03-22 13:30:00',

'Loss', 1, 'Jungle', 'Amumu', 'Classic');

INSERT INTO MatchHistory VALUES ('Mattress World', TIMESTAMP '2018-03-24 17:00:00', TIMESTAMP '2018-03-22 17:40:00',

'Win', 1, 'Middle', 'Fizz', 'Super Galaxy');

INSERT INTO MatchHistory VALUES ('Key Key', TIMESTAMP '2018-03-25 03:00:00', TIMESTAMP '2018-03-22 03:30:00',

'Loss', 2, 'Support', 'Rakan', 'Classic');

SELECT \* FROM Servers;

SELECT \* FROM Accounts;

SELECT \* FROM Products;

SELECT \* FROM Owns;

SELECT \* FROM Heroes;

SELECT \* FROM Skins;

SELECT \* FROM MatchHistory;