NAYANA DAVIS

SQL JOINS

WHAT WE'VE LEARNED TO DO WITH DATABASES SO FAR

- how to connect to a local or remote db
- how to add, remove, edit data
- how to perform simple queries
- how to aggregate, group and sort data

SQL COMMANDS

- SELECT
- CREATE
- INSERT
- DELETE
- UPDATE
- ORDER BY
- HAVING
- LIKE
- DISTINCT
- LIMIT

NORMALIZED VS DENORMALIZED DATA

- Normalized structures have a single table per entity, and use many foreign keys or link tables to connect the entities.
- Denormalized tables have fewer tables and may (for example) place all of the tweets and the information on users in one table.

NORMALIZED VS DENORMALIZED DATA

Normalized

Normalized – Data is broken into multiple tables

Product	
ProductID	Desc
1	Mtn Bike #778
2	Road Bike #123
3	Touring Bike #222

Color	
ColorID	Desc
1	Red
2	Black
3	Silver
4	Mauve

Product-Color	
ProductID	ColorID
1	1
1	2
2	1
2	2
2	3
3	1
3	3
3	4

Denormalized

Denormalized – Data combined

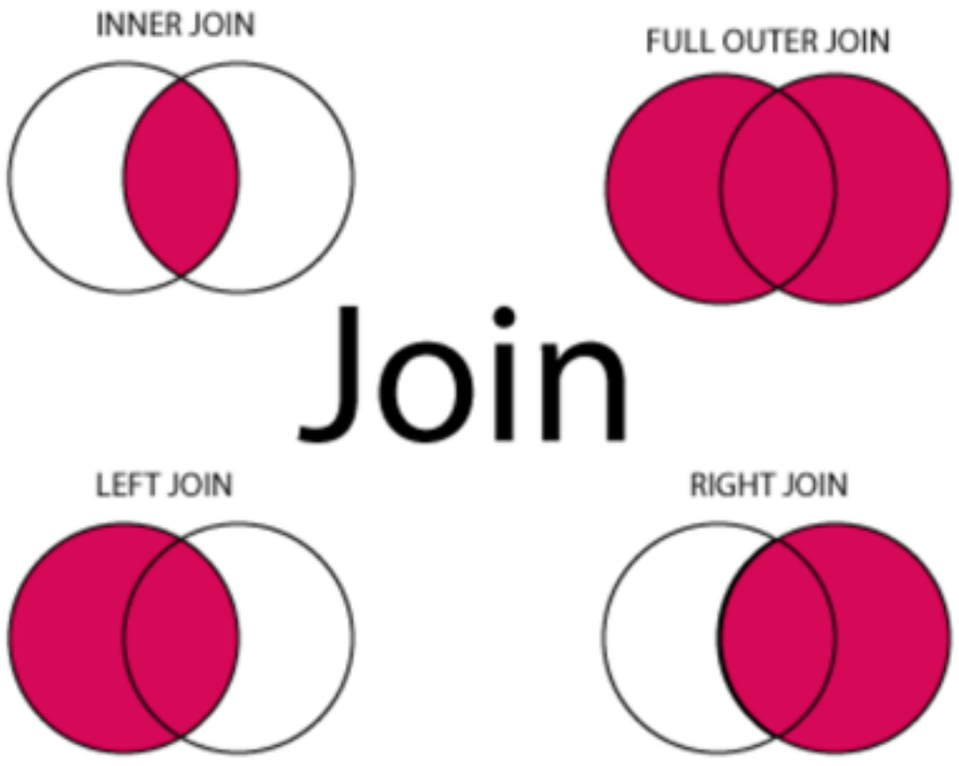
Product	(denormalized)			
ProductSK	ProductID	ColorID	Desc	Color
1	1	1	Mtn Bike #778	Red
2	1	2	Mtn Bike #778	Black
3	2	1	Road Bike #123	Red
4	2	2	Road Bike #123	Black
5	2	3	Road Bike #123	Silver
6	3	1	Touring Bike #222	Red
7	3	3	Touring Bike #222	Silver
8	3	4	Touring Bike #222	Mauve

NORMALIZED VS DENORMALIZED DATA

Normalized tables save the storage space by separating the information. However, if we ever need to access those two pieces of information, we would need to join the two tables, which can be a fairly slow operation.

JOINS

The natural way to merge data in SQL



JOINS

- SQL joins are used when data is spread in different tables. A join operation allows to combine rows from two or more tables in a single new table. In order for this to be possible, a common field between the tables need to exist.

- Join operations can be thought of as operations between two sets, where records with the same key are combined and records missing in one set are either discarded or included as NULL values.

TEXT

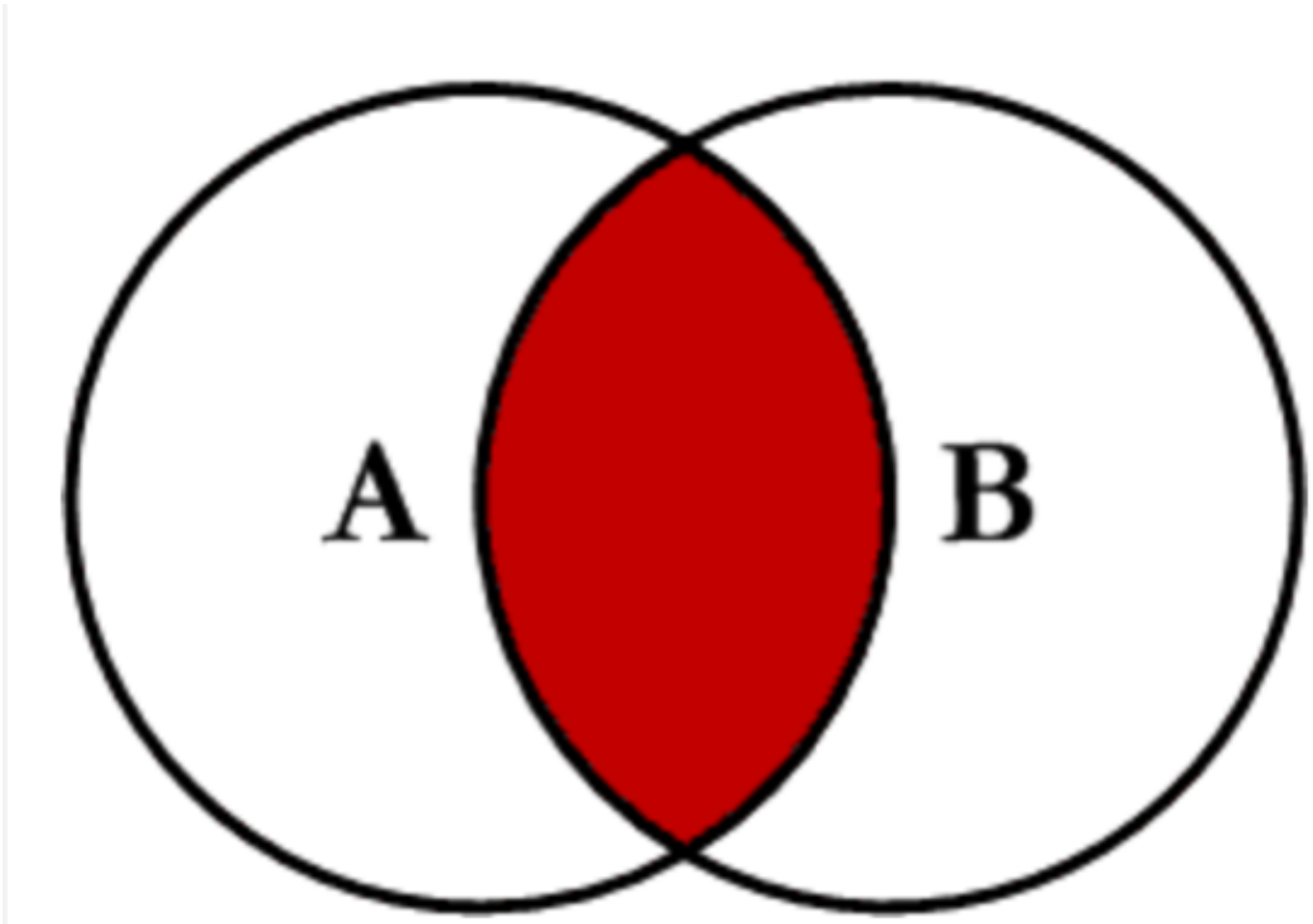
FOREIGN KEY VS PRIMARY KEY

Primary Key	Foreign Key
Primary key uniquely identify a record in the table.	Foreign key is a field in the table that is primary key in another table.
Primary Key can't accept null values.	Foreign key can accept multiple null value.
By default, Primary key is clustered index and data in the database table is physically organized in the sequence of clustered index.	Foreign key do not automatically create an index, clustered or non-clustered. You can manually create an index on foreign key.
We can have only one Primary key in a table.	We can have more than one foreign key in a table.

Note: clustered index means it impacts the way records are stored in a table

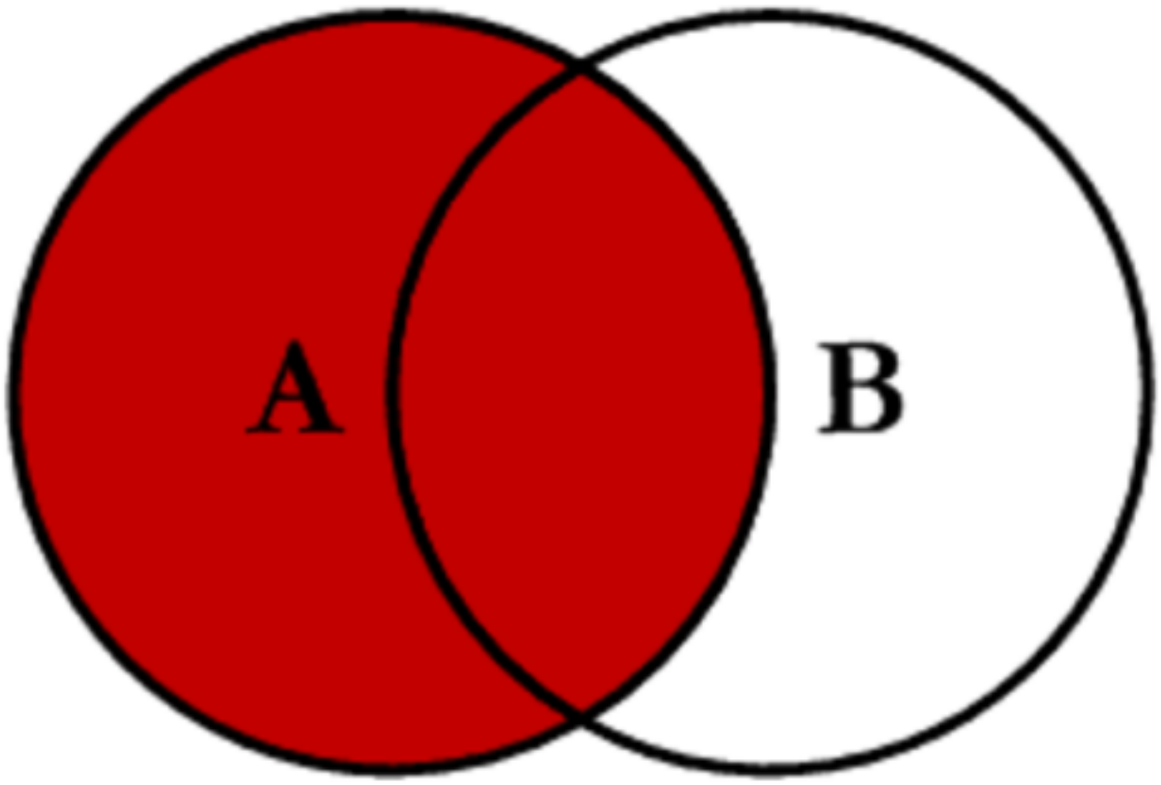
INNER JOIN

This is the simplest, most understood Join and is the most common. This query will return all of the records in the left table (table A) that have a matching record in the right table (table B).



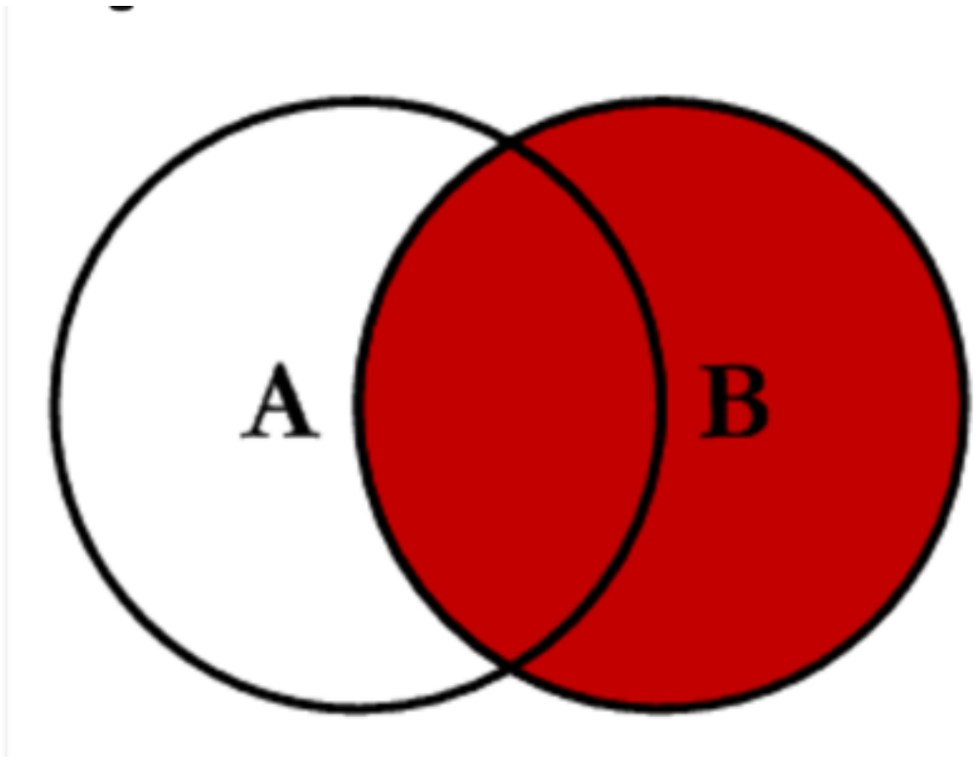
LEFT JOIN

This query will return all of the records in the left table (table A) regardless if any of those records have a match in the right table (table B). It will also return any matching records from the right table.



RIGHT JOIN

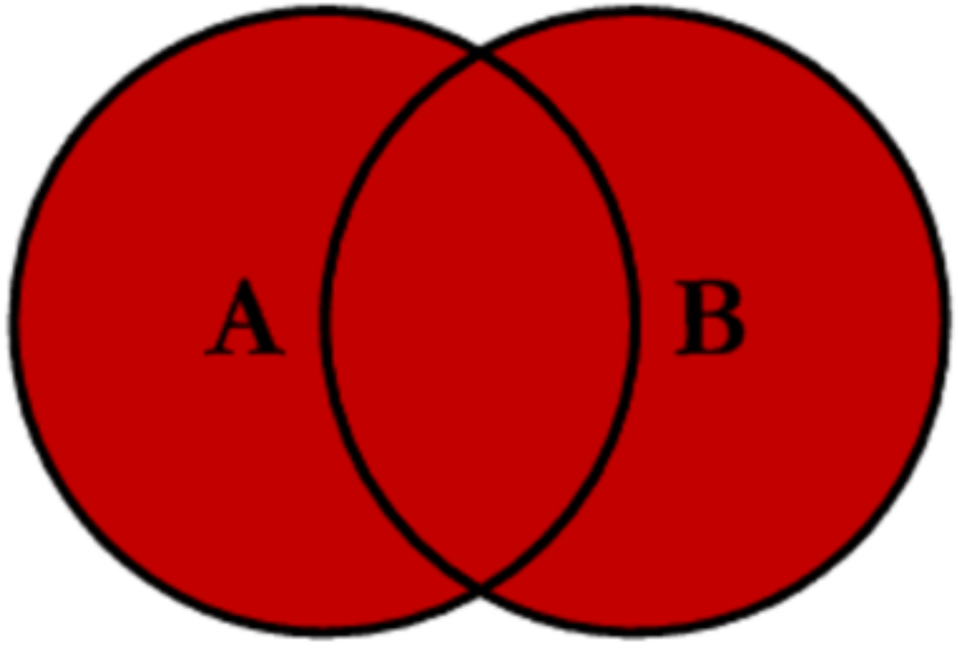
This query will return all of the records in the right table (table B) regardless if any of those records have a match in the left table (table A). It will also return any matching records from the left table.



TEXT

FULL JOIN/OUTER JOIN/FULL OUTER JOIN

This Join can also be referred to as a FULL OUTER JOIN or a FULL JOIN. This query will return all of the records from both tables, joining records from the left table (table A) that match records from the right table (table B)



DEMO: INNER JOIN

Let’s connect to the Northwind database

psql -h dsi.c20gkj5cvu3l.us-east-1.rds.amazonaws.com -p 5432 -U dsi_student northwind

password: gastudents

TEXT

DEMO: INNER JOIN

Let's look at all our tables in the database and then check out the orders and customers tables in full

\dt

TABLE customers;

TABLE orders;

TEXT

DEMO: INNER JOIN

Let's consider a few columns of the orders table:

OrderID	CustomerID	OrderDate
10308	2	1996-09-18
10309	37	1996-09-19
10310	77	1996-09-20

And a few in customers:

CustomerID	CompanyName	ContactName	Country
1	Alfreds Futterkiste	Maria Anders	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mexico

DEMO: INNER JOIN

Notice that the CustomerID column in the Orders table refers to the CustomerID in the Customers table. The relationship between the two tables from before is the CustomerID column. We can thus JOIN the two tables in order to obtain a table like the following, where the information contained in the two tables is joined in a single table, using the common key of CustomerID.

OrderID	CompanyName	OrderDate
10308	Ana Trujillo Emparedados y helados	9/18/1996
10365	Antonio Moreno Taquería	11/27/1996
10383	Around the Horn	12/16/1996
10355	Around the Horn	11/15/1996
10278	Berglunds snabbköp	8/12/1996

DEMO: INNER JOIN

```
SELECT <select_list>
FROM Table_A A
INNER JOIN Table_B B
ON A.Key = B.Key
```

```
SELECT orders."OrderID", customers."CompanyName", orders."OrderDate"
FROM orders
INNER JOIN customers
ON orders."CustomerID"=customers."CustomerID";
```

TEXT

YOU DO: LEFT JOIN (3 MINUTES)

Syntax:

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name=table2.column_name;
```

Perform a left join with orders as table 1, customers as table 2

Slack query in students channel

TEXT

YOU DO: RIGHT JOIN (3 MINUTES)

Syntax:

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name=table2.column_name;
```

Perform a right join with orders as table 1, customers as table 2

Slack query in students channel

YOU DO: FULL JOIN (3 MINUTES)

Syntax:

```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name=table2.column_name;
```

Perform a full join with orders as table 1, customers as table 2

Slack query in students channel

TEXT

YOU DO: WRITE QUERIES TO ANSWER THESE QUESTIONS(15 MINUTES)

Look over what you previously learned about SQL:

https://github.com/ga-students/DSI-DC-1/tree/master/week-05

QUESTIONS

Answer these questions:

How many products per category does the catalog contain? Print the answer with the CategoryName, and Count.

What 5 customers are generating the highest revenue? Print a table with CustomerID and Total Revenue. You will need to use data from 3 tables.

In which country are the top 5 suppliers by number of units supplied? Print a table with the supplier's CompanyName, Country and total number of units supplied.

Slack your code in the students channel

SUBQUERIES

- A Subquery or Inner query or Nested query is a query within another SQL query. It is used to further restrict the data to be retrieved by returning data that will be used in the main query as a condition.

- Subqueries can be used with the SELECT, INSERT, UPDATE, and DELETE statements along with the operators like =, <, >, >=, <=, IN, BETWEEN etc.

TEXT

YOU DO: SUBQUERIES (5 MINUTES)

Syntax:

```
SELECT column_name1
    FROM table_name1
    WHERE column_name2 [Comparison Operator]
        (SELECT column_name3
        FROM table_name2
        WHERE condition);
```

Perform a query that uses a subquery that extract all the orders from customers based in France.

Do this again, but this time use a JOIN operation

INDEPENDENT PRACTICE (10 MINUTES)

- Work in pairs: go to http://www.w3schools.com/sql and choose a command you have not hear of. Read about it for 5 minutes, then explain it to your pair (take 2.5 minutes turns).
- Use the last 5 minutes to share some interesting finds with the rest of the class.