Week 2 Day 3

SQL Normalization

SQL: Normalization

REVATURE

Reduce redundancy and improve intregrity

- First Normal Form (1NF)
  - Primary Key
  - No Duplicates
  - Atomic
- Second Normal Form (2NF)
  - 1NF
  - No Partial Dependencies
- Third Normal Form (3NF)
  - 2NF
  - No Transitive Dependencies

SQL: Multiplicity

REVATURE

- The relationship between tables
- One-to-many
  - One entity is related to many in another table
- Many-to-one
  - Opposite of one-to-many
- Many-to-many
  - Many entities are related to many entities in another table
- One-to-one
  - Direct mapping between tables

Multiplicity: Referential Integrity

REVATURE

- When a reference is created you must insure it is not broken
  - If the parent record is deleted, you must:
    - Update the foreign key reference
    - Remove the associated entities with the deleted item
- The easiest way to achieve referential integrity is CASCADE DELETE constraint on the column

SQL: Views

REVATURE

Virtualized tables from DQL statements

- Stores the query results into a virtual table
- Denormalizes tables
- Allows users access to parts of a table
- Hides complexities
- Virtual table will update with the real tables

REVATURE

Normalizing our database

SQL: Aggregate Functions

REVATURE

- Combine the values of multiple rows into a single result
  - MAX(column)
  - MIN(column)
  - AVG(column)
  - SUM(column)
- Used with SELECT clause
- Must use GROUP BY when SELECTING multiple columns
- Use inside of a HAVING clause

SQL: Scalar Functions

REVATURE

- Operate on singular rows, typically transforming the result
  - LENGTH(column)
  - UPPER(column)
  - LOWER(column)
  - ABS(column)
- Can be used with SELECT and WHERE clauses

SQL: Subqueries

REVATURE

- Query in place of any DML statement
  - Outer query uses the result of the inner
  - Often used with Joins

Code

```
SELECT name, listed_price
FROM paintings
WHERE listed_price > (
    SELECT AVG(listed_price)
    FROM paintings
);
```

Code

```
SELECT
  first_name,
  last_name,
  (
    SELECT count(*) AS paintings
    FROM sales
    WHERE collectors.id = sales.collector_id
  )
FROM collectors;
```

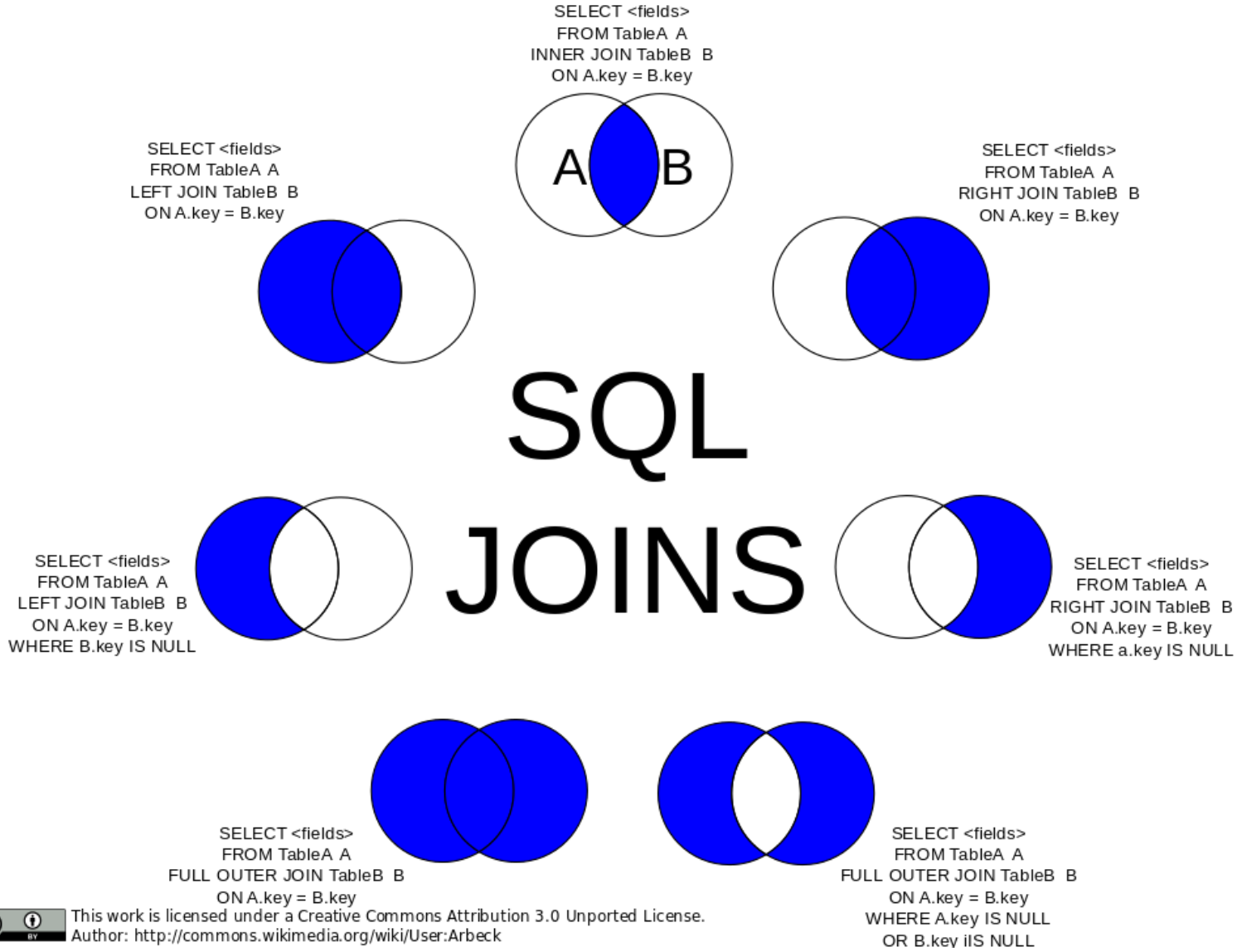
SQL: Joins

REVATURE

Another way to denomoralize tables by combining them

- INNER JOIN
  - Returns the rows that match between the tables
- OUTER/LEFT JOIN
  - Returns matching rows and null of left
- OUTER/RIGHT JOIN
  - Returns matching rows and null of right
- OUTER/FULL JOIN
  - Returns all rows
- SELF JOIN
  - Inner join on columns of the same table
- CROSS JOIN
  - Returns cartesian product
- Natural
  - Auto join

SQL: Joins

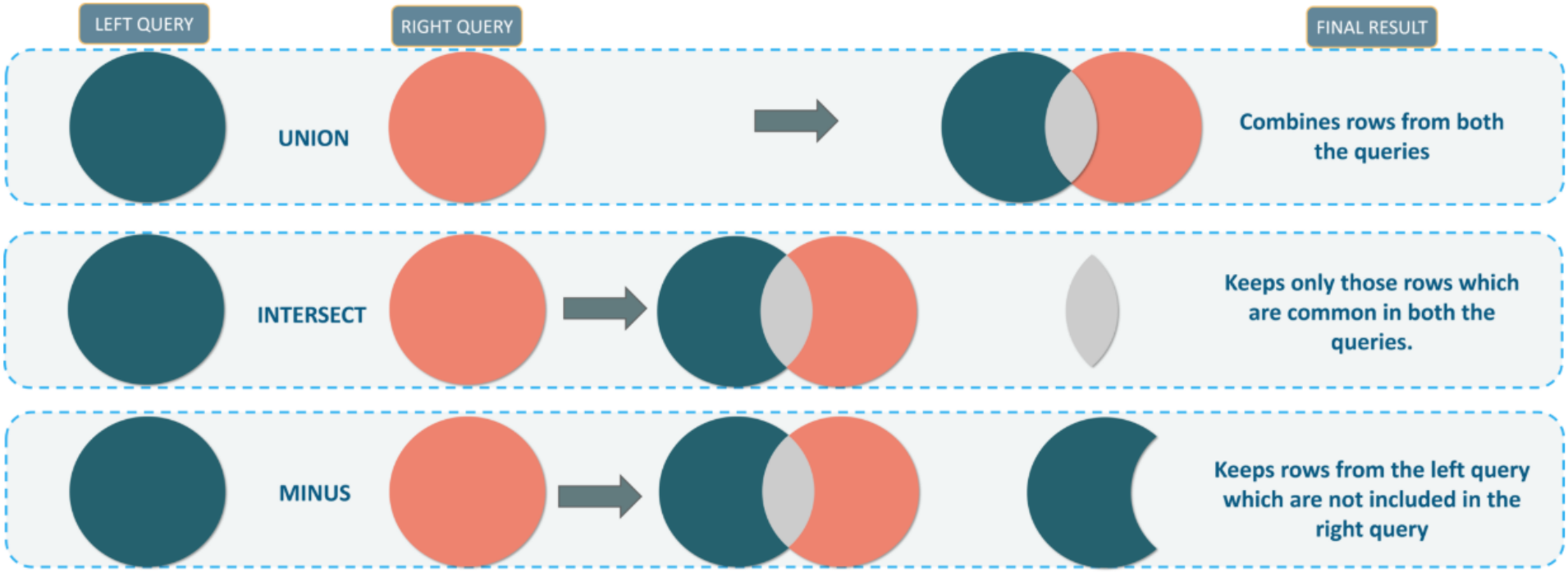


SQL: Set Operators

REVATURE

- Like JOINS, but join result sets from SELECTS
- UNION
  - Returns every column between the sets
- INTERSECT
  - Returns the sets have in common
- MINUS
  - Removes elements from first set which are present in the second set
- EXCEPT
  - The opposite of MINUS

SQL: Set Operators



https://www.google.com/url?sa=i&url=https%3A%2F%2Fvisualizeright.com%2F2019%2F03%2F15%2Fset-operators%2F&psig=AOvVaw3TlzVVx2LtjtrNkivU7VYK&ust=1665085654991000&source=images&cd=vfe&ved=0CAwQjRxqFwoTCKiIuMztyfoCFQAAAAAdAAAAABAN

REVATURE

Advanced SQL with Chinook DB