

# **COMP 353 - Databases**

**3rd Session of Lab XJ**

**Omid Reza Heidari**

**2024-02-05**

# Primary Key

A primary key is a special field in the relational table that is used to identify each record

## Main Features:

- Not null
- Unique

## Examples:

Creating a customers table with a unique id

Creating a passengers table with a unique passport number

# Foreign Key

A foreign key is a key used to link two tables together

It is field/collection of fields in one table that refers to the primary key in another table

Different behaviours in foreign key modification:

- Restricted
- No Action
- Cascade

Create two tables : orders, items with the indicated behaviours

# Where conditions

We can use more keywords in the where condition to build more complex queries

And | OR | NOT

- The AND operator displays a record if all the conditions separated by AND are TRUE.
- The OR operator displays a record if any of the conditions separated by OR is TRUE.
- The NOT operator displays a record if the condition(s) is NOT TRUE.

Subqueries (will be discussed in the coming sessions!)

# Joins

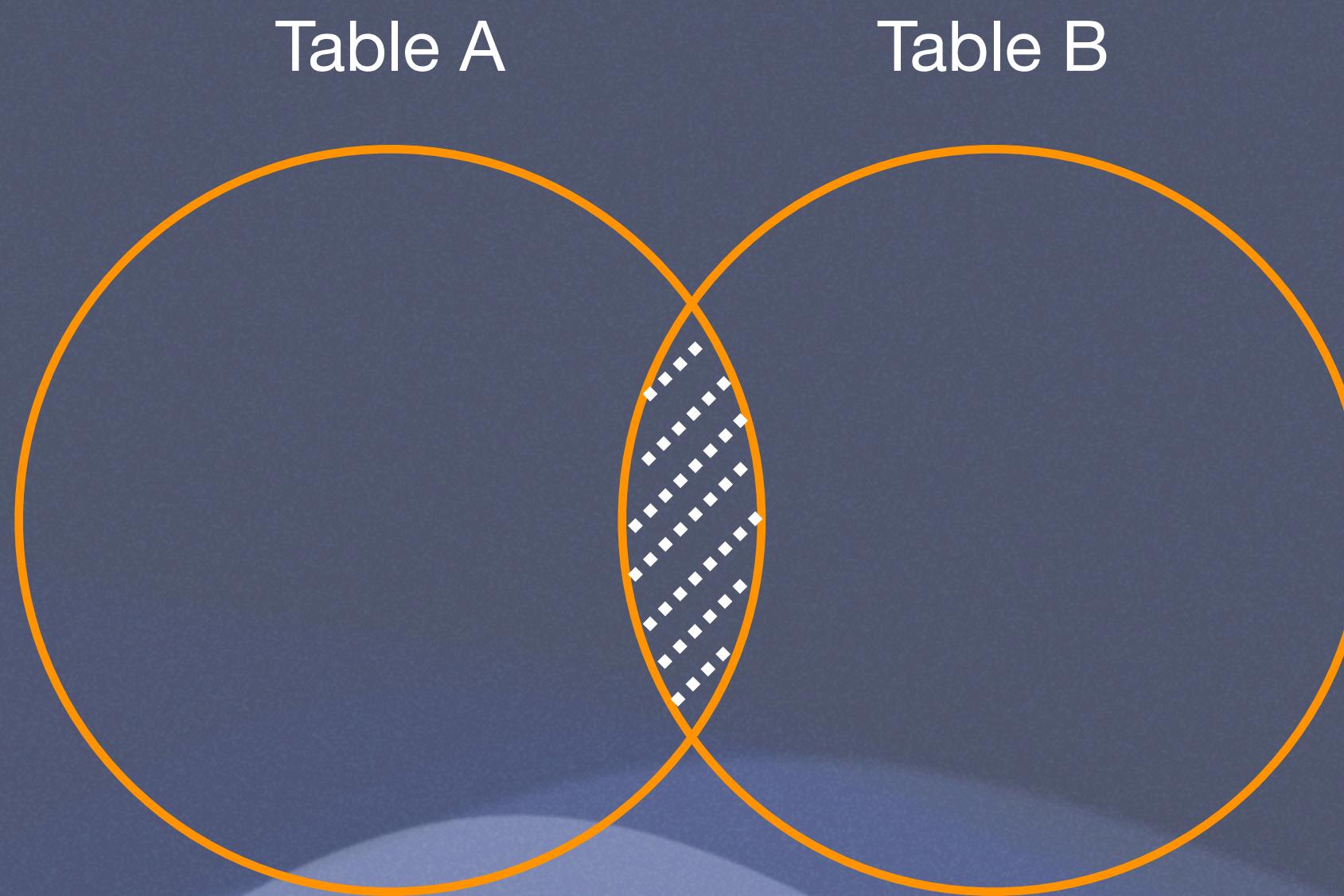
**Joins are used to retrieve data from two or more data tables, based on a related column between them.**

Different types of join:

- Inner join
- Full join
- Left join
- Right join
- Self Join

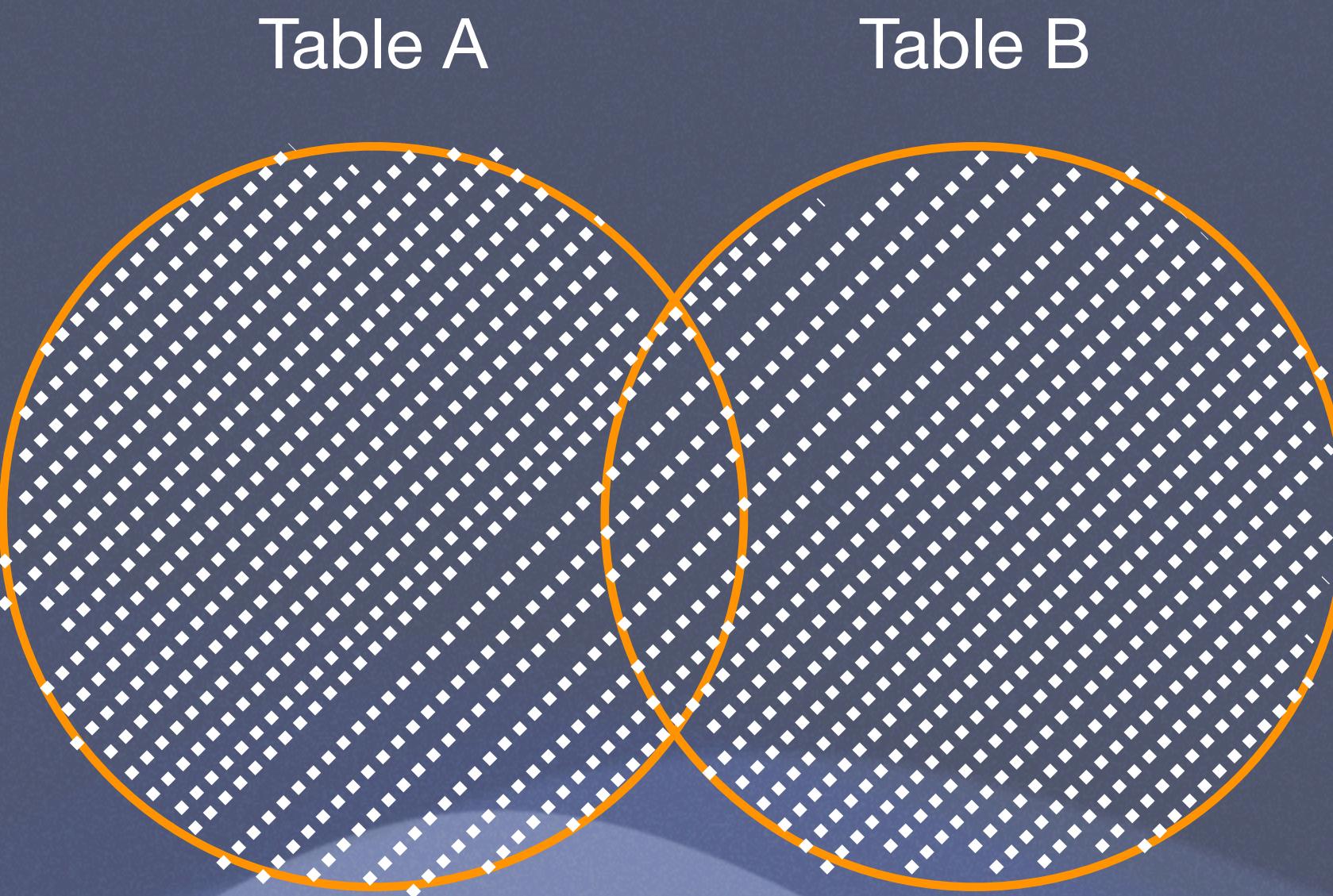
# Inner Join

The INNER JOIN command selects records that have matching values in both tables.



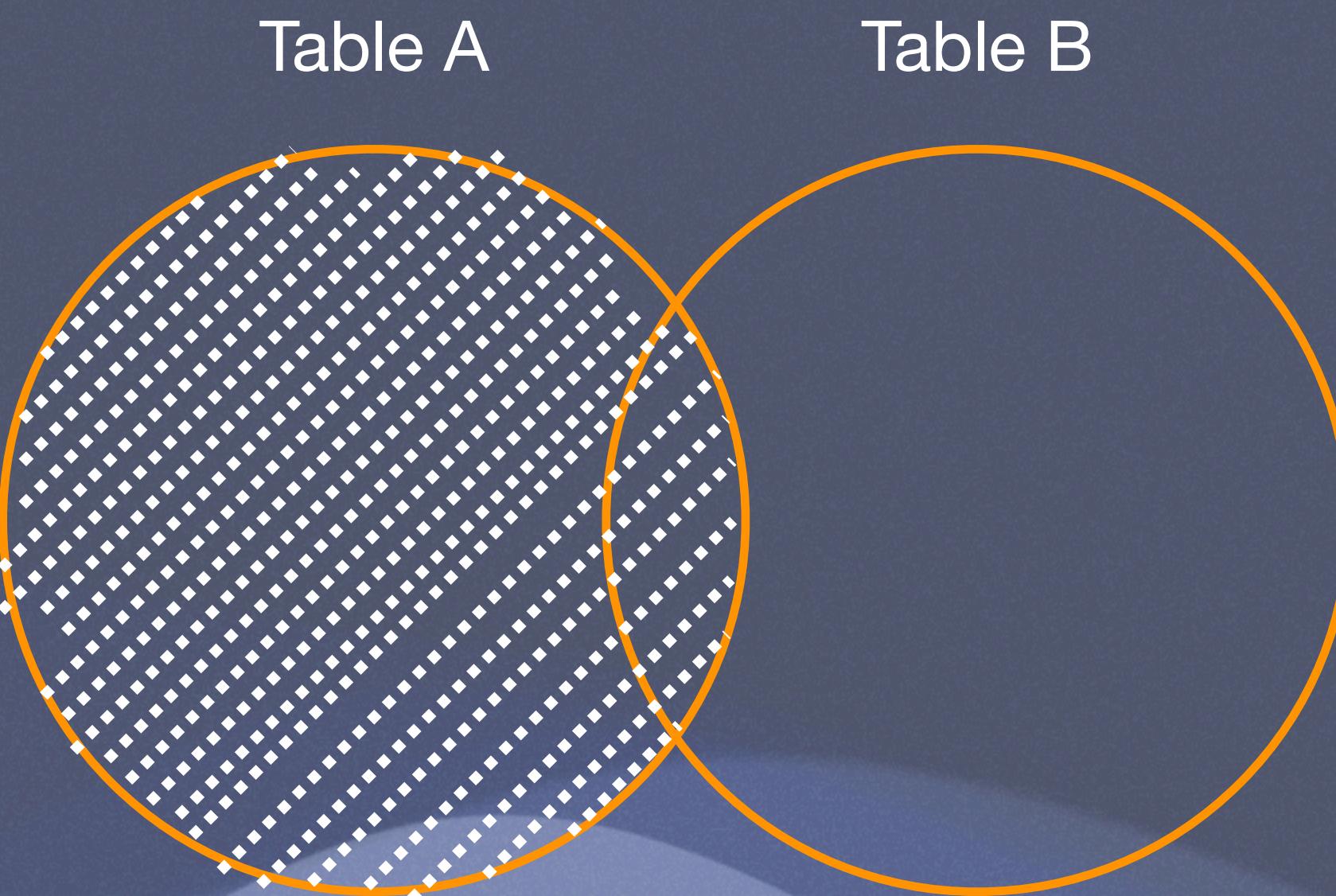
# Full Join

The FULL OUTER JOIN keyword returns all records when there is a match in left or right table records.



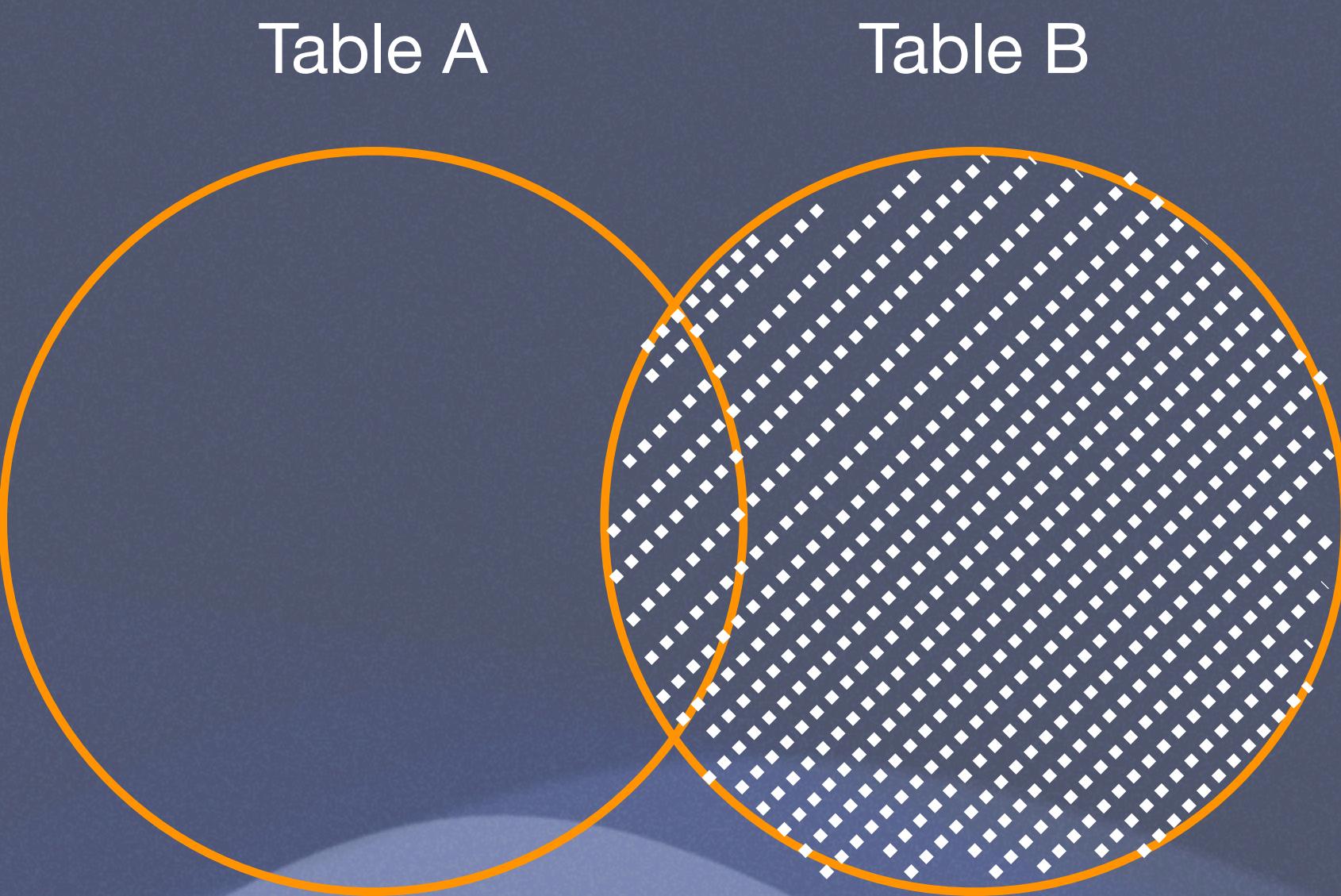
# Left Join

The LEFT JOIN keyword returns all records from the left table, and the matching records from the right table. The result is 0 records from the right side, if there is no match.



# Right Join

The RIGHT JOIN keyword returns all records from the right table, and the matching records from the left table. The result is 0 records from the left side, if there is no match.



# Self Join

A self join is a regular join, but the table is joined with itself.

Example:

Find the users that are from the same country?

# Null

A field with a **NUL** value has no value

Comparison operators cannot be used for Null values. What we should do in the queries for this values?

Two operators are available for it:

- IS NULL
- IS NOT NULL

More functions will be discussed in the coming sessions!

# Order By

Used to sort the result-set from a **SELECT** statement in ascending or descending order.

Different types of sorting in MySQL:

- Descending
- Ascending

Note: The default mode for sorting is ascending.

# Data Types

**There are 3 main data types: string, numeric, data and time**

There are 3 main data types:

- string
- Numeric
- data and time

Note: The following data types are supported in MySQL version 8.0 and might be different in different versions.

# String

- CHAR(size): Fixed length string (size could be from 0 to 255 - default=1)
- VARCHAR(size): Variable length string (size could be from 0 to 65535)
- BINARY(size): Equal to CHAR but stores binary byte strings.
- VARBINARY(size): Equal to VARCHAR but stores binary byte strings.
- TINYBLOB: For BLOBS (Binary Large OBjects). Max length: 255 bytes
- TINYTEXT: Holds a string with a maximum length of 255 characters
- TEXT(size): Holds a string with a maximum length of 65535 characters

# String

- BLOB(size): For BLOBs (Binary Large OBjects). Holds up to 65535 bytes of data.
- MEDIUMTEXT: Holds a string with a maximum length of 16777215 characters.
- MEDIUMBLOB: For BLOBs (Binary Large OBjects). Holds up to 16777215 bytes of data.
- LONGTEXT: Holds a string with a maximum length of 4294967295 characters.
- LONGBLOB: For BLOBs (Binary Large OBjects). Holds up to 4294967,295 bytes of data.
- ENUM(val1, val2, val3, ...): A string object that can have only one value, chosen from a list of possible values. You can list up to 65535 values in an ENUM list.
- SET(val1, val2, val3, ...): A string object that can have 0 or more values, chosen from a list of possible values. You can list up to 64 values in a SET list.

# Numeric

- BIT(size): A bit-value type. The number of bits per value is specified in size. The size parameter can hold a value from 1 to 64 (Default = 1).
- TINYINT(size): A very small integer. Signed range is from -128 to 127. Unsigned range is from 0 to 255.
- BOOL: 0=False, otherwise=True
- BOOLEAN: Equal to BOOL
- SMALLINT(size): A small integer. Signed range is from -32768 to 32767. Unsigned range is from 0 to 65535.
- MEDIUMINT(size): A medium integer. Signed range is from -8388608 to 8388607. Unsigned range is from 0 to 16777215
- INT(size): A medium integer. Signed range is from -2147483648 to 2147483647. Unsigned range is from 0 to 4294967295.
- INTEGER(size) Equal to INT(size)

# Numeric

- **BIGINT(size)**: A large integer. Signed range is from -9223372036854775808 to 9223372036854775807. Unsigned range is from 0 to 18446744073709551615.
- **FLOAT(size, d)**: A floating point number. The total number of digits is specified in size. The number of digits after the decimal point is specified in the d parameter. Note: deprecated in Version 8.0.17
- **FLOAT(p)**: A floating point number. MySQL uses the p value to determine whether to use FLOAT or DOUBLE for the resulting data type. If p is from 0 to 24, the data type becomes FLOAT(). If p is from 25 to 53, the data type becomes DOUBLE()
- **DOUBLE(size, d)**: A normal-size floating point number. The total number of digits is specified in size. The number of digits after the decimal point is specified in the d parameter.
- **DOUBLE PRECISION(size, d)**
- **DECIMAL(size, d)**: An exact fixed-point number. The total number of digits is specified in size (max value = 65, default=10). The number of digits after the decimal point is specified in the d parameter (max value=30, default =0)
- **DEC(size, d)**: Equal to DECIMAL(size,d)

# Date and Time

- DATE: A date. Format: YYYY-MM-DD.
- DATETIME(fsp): A date and time combination. Format: YYYY-MM-DD hh:mm:ss.
- TIMESTAMP(fsp): A timestamp. TIMESTAMP values are stored as the number of seconds since the Unix epoch ('1970-01-01 00:00:00' UTC). Format: YYYY-MM-DD hh:mm:ss.
- TIME(fsp): A time. Format: hh:mm:ss.
- YEAR: A year in four-digit format. Values allowed in four-digit format: 1901 to 2155, and 0000.

# Assignment

Create the following tables with SQL queries

Actors

Id	Int
Name	varchar(50)
Gender	char(1)

Directors

Id	Int
Name	varchar(50)

actor\_movie

actor_id	Int
movie_id	varchar(50)

Movies

Id	Int
Title	varchar(50)
Published_at	Year
duration_in_sec	Int
director_id	Int

# Assignment

**Insert some data to the tables & run the below queries.**

- Find movies made during World War II (1939-1945).
- Find the average runtime of the movies.
- Find the longest movie before 1980.
- Find movies which Humphrey Bogart acted in them.
- List movies along with their director names.
- Find movies directed by Billy Wilder.
- Find the director of the earliest movie.
- Find movies with female actors (actresses).
- Find directors who directed at least two movies released before 1960.
- List actors who have appeared in more than one movie, along with the count of movies they acted in.
- Find the decade with the highest average runtime of movies.
- Find the directors who have directed movies with an average runtime greater than 150 minutes.

# **Any Question?**

**See you in the next week :)**