

# SQL

Martina Cvinčková



# AGENDA

- **Opakování**
- **Databázový jazyk SQL**
  - **Spojování tabulek (*SQL join*)**

# SQL

## Opakování

# AGREGAČNÍ FUNKCE

V dotazu umožní spočítat výsledek dle zvolené funkce.

**SELECT**

**AVG**(sloupec), **sloupec2**

**FROM**

tabulka

**GROUP BY**

**sloupec2;**

# AGREGAČNÍ FUNKCE

**AVG**(sloupec)

Průměr hodnot sloupce

**MAX**(sloupec)

Maximální hodnota sloupce

**MIN**(sloupec)

Minimální hodnota sloupce

**SUM**(sloupec)

Součet hodnot sloupce

**COUNT**(sloupec)

Počet hodnot v tabulce

# HAVING

Umožňuje vytvořit podmínku pro výsledek agregační funkce.

```
SELECT SUM(sloupec), sloupec2  
  FROM tabulka  
  
GROUP BY sloupec2  
  
HAVING SUM(sloupec) > 0;
```

# Základní syntaxe dotazu SQL – SELECT

## **SELECT**

seznam sloupců, agregace

## **FROM**

zdrojové tabulky

## **WHERE**

podmínky

## **GROUP BY**

seznam sloupců

## **HAVING**

podmínky pro výsledek agregace

## **ORDER BY**

seznam sloupců

## **LIMIT**

počet řádků;

## DOTAZY

- Ze kterých 3 produktů se prodalo nejvíc kusů?  
Zajímají nás pouze výrobky, kterých jsme prodali více jak 10 000 ks a mají ID menší než 500.



...

```
SELECT
    ProductID,
    SUM(Units) AS UnitSum
FROM
    Sales
WHERE
    ProductID < 500
GROUP BY
    ProductID
HAVING
    UnitSum > 10000
ORDER BY
    UnitSum DESC
LIMIT
    3;
```

## DOTAZY

- Kolik produktů levnějších než 200\$ je nabízeno v každé kategorii? Zobrazte pouze ty kategorie, které mají alespoň 35 produktů a výsledek seřadte sestupně podle počtu produktů.

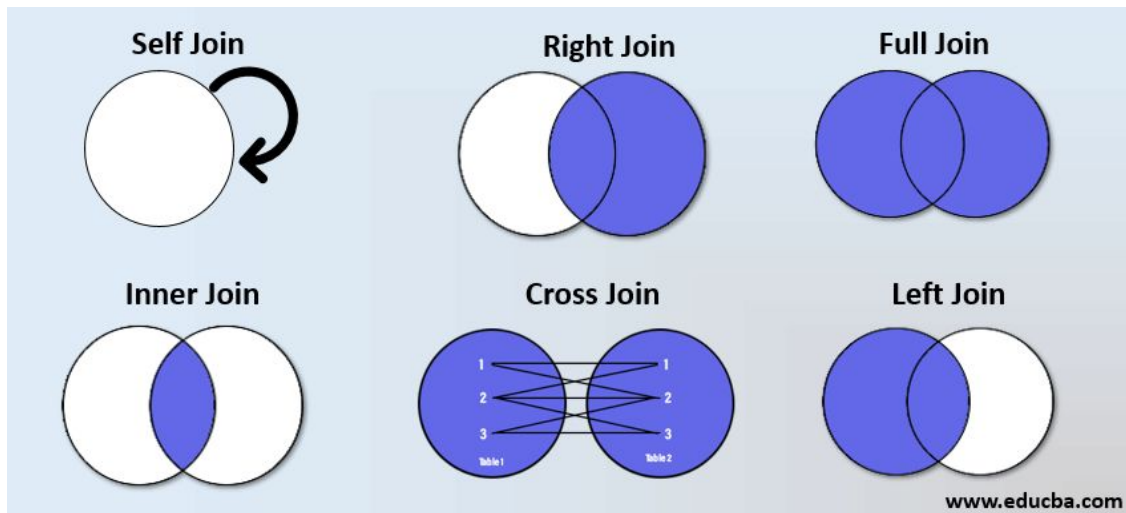
...

```
SELECT
    Category,
    COUNT(productid) AS ProductCount
FROM
    product
WHERE
    PriceNew < 200
GROUP BY
    Category
HAVING
    ProductCount >= 35
ORDER BY
    ProductCount DESC
```

# SQL

Spojování tabulek (joins) – inner, left, right, full

# TYPY SPOJENÍ TABULEK



- Vnitřní spojení – (INNER) JOIN
- Vnější spojení – LEFT/RIGHT/FULL (OUTER) JOIN
- Kartézské spojení – CROSS JOIN

# SQL

## INNER JOIN

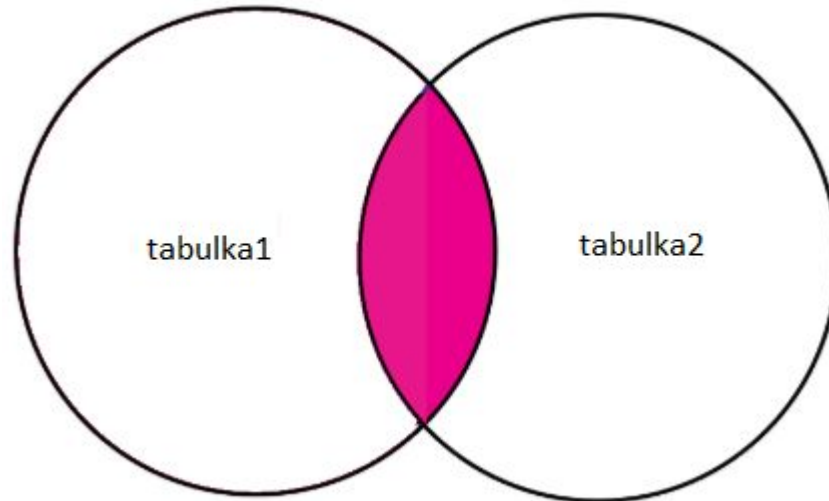
# INNER JOIN

## SELECT

tabulka1.sloupec1, tabulka1.sloupec2, tabulka2.sloupec1

## FROM

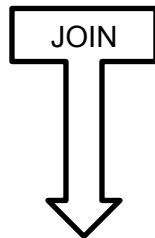
tabulka1 **INNER JOIN** tabulka2 **ON** tabulka1.sloupec1 = tabulka2.sloupec1;



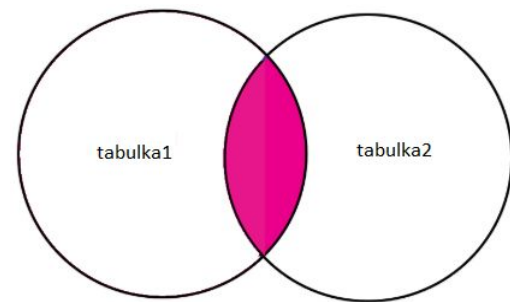
# INNER JOIN

Spojí řádky dvou tabulek na základě podmínky ON:

ProductID	Zip	Units	Revenue
449	29485	1	1039.448
656	06716	1	913.4475
792	46310	1	80.85
2218	19559	1	167.475



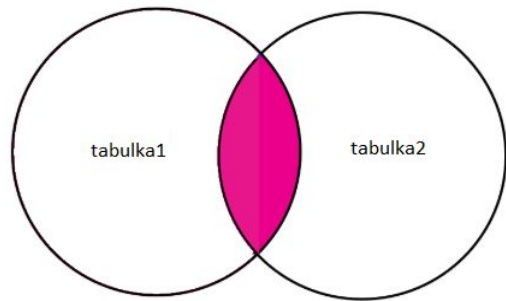
Zip	City	State
29204	Columbia	SC
29485	Summerville	SC
19559	Strausstown	PA
46310	Demotte	IN



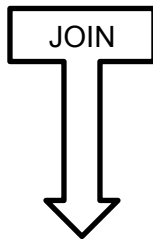


# INNER JOIN

Spojí řádky dvou tabulek na základě podmínky ON:



ProductID	Zip	Units	Revenue
449	29485	1	1039.448
656	06716	1	913.4475
792	46310	1	80.85
2218	19559	1	167.475



Zip	City	State
29204	Columbia	SC
29485	Summerville	SC
19559	Strausstown	PA
46310	Demotte	IN

ProductID	Zip	Units	Revenue	Zip	City	State
449	29485	1	1039.448	29485	Summerville	SC
792	46310	1	80.85	46310	Demotte	IN
2218	19559	1	167.475	19559	Strausstown	PA

# INNER JOIN

**INNER JOIN** a **JOIN** – ekvivalentní zápis

**SELECT**

tabulka1.sloupec1, tabulka1.sloupec2, tabulka2.sloupec1

**FROM**

tabulka1 **INNER JOIN** tabulka2 **ON** tabulka1.sloupec1 = tabulka2.sloupec1;

tabulka1 **JOIN** tabulka2 **ON** tabulka1.sloupec1 = tabulka2.sloupec1;

# ÚKOLY – INNER JOIN

- Vyberte základní informace o produktech (productID a product) a připojte k nim základní informace o výrobci (manufacturerID a manufacturer).

# ÚKOLY – INNER JOIN

- Vyberte základní informace o produktech (productID a product) a připojte k nim základní informace o výrobci (manufacturerID a manufacturer).

SELECT

productid,  
product,  
product.manufacturerid,  
manufacturer

FROM

product JOIN manufacturer ON product.ManufacturerID =  
manufacturer.ManufacturerID

# ÚKOLY – INNER JOIN

- Vyberte TOP 10 nejvýnosnějších prodejů dle příjmů a přidejte k nim odpovídající informace z tabulky Country

# ÚKOLY – INNER JOIN

- Vyberte TOP 10 nejvýnosnějších prodejů dle příjmů a přidejte k nim odpovídající informace z tabulky Country.

```
SELECT *  
FROM  
    Sales s JOIN Country c ON s.Zip = c.Zip  
ORDER BY  
    revenue desc  
LIMIT 10;
```

Samostatná práce 1

Inner Join

# SQL

## LEFT JOIN



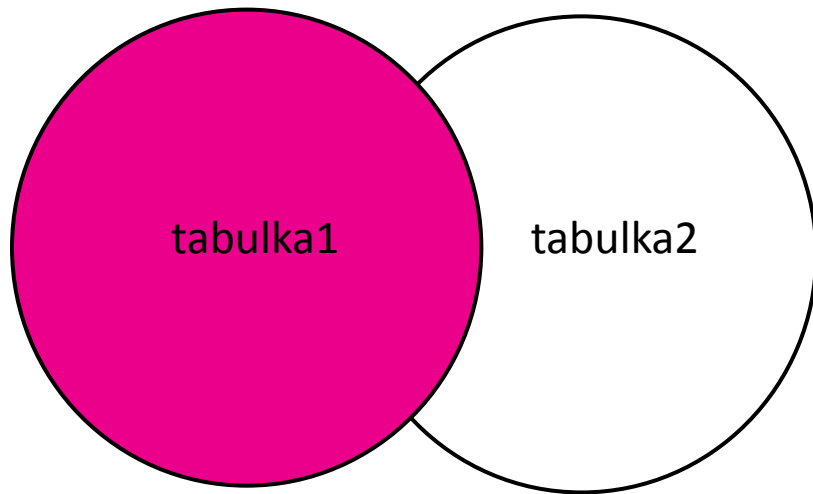
# LEFT JOIN

## SELECT

tabulka1.sloupec1, tabulka1.sloupec2, tabulka2.sloupec1

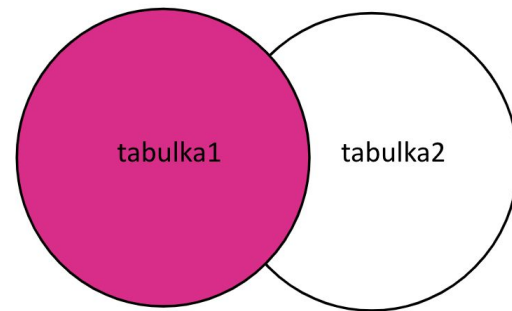
## FROM

tabulka1 **LEFT JOIN** tabulka2 **ON** tabulka1.sloupec1 = tabulka2.sloupec1;



# LEFT JOIN

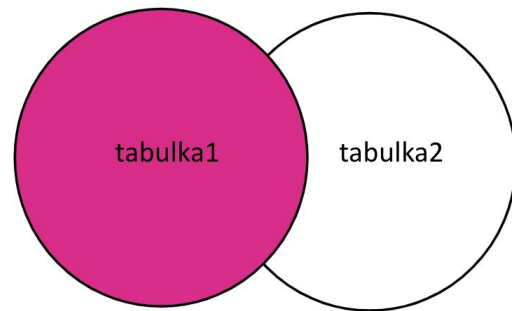
Spojí řádky dvou tabulek na základě podmínky ON:



ProductID	Zip	Units	Revenue	LEFT JOIN	Zip	City	State
449	29485	1	1039.448		29204	Columbia	SC
656	06716	1	913.4475		29485	Summerville	SC
792	46310	1	80.85		19559	Strausstown	PA
2218	19559	1	167.475		46310	Demotte	IN

# LEFT JOIN

Spojí řádky dvou tabulek na základě podmínky ON:



ProductID	Zip	Units	Revenue	LEFT JOIN		
449	29485	1	1039.448	29204	Columbia	SC
656	06716	1	913.4475	29485	Summerville	SC
792	46310	1	80.85	19559	Strausstown	PA
2218	19559	1	167.475	46310	Demotte	IN

ProductID	Zip	Units	Revenue	Zip	City	State
449	29485	1	1039.448	29485	Summerville	SC
656	06716	1	913.4475	NULL	NULL	NULL
792	46310	1	80.85	46310	Demotte	IN
2218	19559	1	167.475	19559	Strausstown	PA

# ÚKOLY – LEFT JOIN

- Kolik různorodých produktů se prodalo v každém z měst? Zobraz i města, kde se neprodal žádný výrobek.

# ÚKOLY – LEFT JOIN

- Kolik různorodých produktů se prodalo v každém z měst? Zobraz i města, kde se neprodal žádný výrobek.

```
SELECT c.CITY,  
       COUNT(DISTINCT s.ProductID) pocetProduktu  
FROM country c  
     LEFT JOIN  
     sales s ON s.ZIP = c.ZIP  
GROUP BY  
     c.City  
ORDER BY  
     pocetProduktu;
```

Samostatná práce 2

Inner Join & Left Join

Q&A