

# Databázové systémy

JOIN

# Motivácia - normalizácia databáz

- **Normalizácia databáz** je systematický proces návrhu relačných databáz, ktorého cieľom je minimalizovať redundanciu (nadbytočnosť) dát a eliminovať anomálie pri modifikácii dát (pri vkladaní, aktualizácii a mazaní).
- **Zjednodušené povedané:** Je to technika, pomocou ktorej zložité a veľké tabuľky dekomponujeme (rozdeľujeme) na menšie, logicky prepojené tabuľky tak, aby sa každá informácia nachádzala v databáze práve na jednom jedinom mieste.

# OsobaDB

SELECT \* FROM Osoba -- 23 riadkov

| id | meno     | priezvisko | rodne_priezvisko | dat_nar    | dat_smrti  | pohlavie | vyska | vaha  | otec | matka |
|----|----------|------------|------------------|------------|------------|----------|-------|-------|------|-------|
| 1  | Adam     | Prvý       | NULL             | 1918-05-11 | 1968-10-01 | m        | 180.0 | 80.0  | NULL | NULL  |
| 2  | Eva      | Prvá       | Druhá            | 1919-01-09 | 1988-07-22 | z        | 160.0 | 60.0  | NULL | NULL  |
| 3  | Zoly     | Múdry      | NULL             | 1918-04-07 | 1990-09-23 | m        | 175.5 | 75.0  | NULL | NULL  |
| 4  | Nasa     | Kováčová   | Rostová          | 1928-02-05 | 1965-03-11 | z        | 155.0 | 99.0  | NULL | NULL  |
| 5  | Jozef    | Urban      | NULL             | 1922-10-19 | NULL       | m        | 199.5 | NULL  | NULL | NULL  |
| 6  | Mária    | Urbanová   | Nováková         | 1937-12-08 | NULL       | z        | 172.5 | 57.5  | 1    | 2     |
| 7  | Patrik   | Novák      | Novák            | 1945-06-19 | NULL       | m        | 182.5 | 89.5  | 1    | 2     |
| 8  | Patricia | Nováková   | Halušková        | 1952-01-08 | NULL       | z        | 143.5 | 35.0  | NULL | NULL  |
| 9  | Michal   | Kováč      | Kováč            | 1942-04-10 | NULL       | m        | 167.0 | 88.0  | 3    | 2     |
| 10 | Roman    | Kováč      | Kováč            | 1948-05-20 | NULL       | m        | 179.5 | 78.5  | 3    | 4     |
| 11 | Peter    | Horváth    | Horváth          | 1959-07-02 | 2000-12-31 | m        | 193.0 | 110.5 | NULL | NULL  |
| 12 | Lucia    | Horváthová | Urbanová         | 1959-01-13 | NULL       | z        | 156.5 | 45.5  | 5    | 6     |
| 13 | Urban    | Urban      | Urban            | 1957-03-31 | NULL       | m        | 138.2 | 24.5  | 5    | 6     |
| 14 | Dáša     | Nováková   | Nováková         | 1970-07-17 | NULL       | z        | 167.0 | 55.0  | 7    | 8     |
| 15 | Viera    | Silná      | Nováková         | 1973-02-13 | NULL       | z        | 169.5 | 63.0  | 7    | 8     |
| 16 | Vladimír | Silný      | Silný            | 1974-08-01 | 2002-12-04 | m        | 175.5 | 73.0  | NULL | NULL  |
| 17 | Milena   | Slabá      | Slabá            | 1979-09-14 | NULL       | z        | 164.0 | 64.0  | NULL | NULL  |
| 18 | Ján      | Horváth    | Horváth          | 1982-01-16 | NULL       | m        | 159.5 | 65.5  | 11   | 12    |
| 19 | Zuzana   | Silná      | Silná            | 2002-03-01 | NULL       | z        | 158.5 | 60.0  | 16   | 15    |
| 20 | Zuzana   | Slabá      | Slabá            | 1999-12-16 | NULL       | z        | 171.5 | 54.5  | 16   | 17    |
| 21 | Zuzana   | Pravá      | Pravá            | 1990-11-26 | NULL       | z        | 170.5 | 60.5  | 16   | 17    |
| 22 | Zuzana   | ¼avá       | ¼avá             | 1931-01-14 | NULL       | z        | 195.5 | 58.5  | 16   | 17    |
| 23 | Zuzana   | Stredná    | Stredná          | 1945-04-08 | NULL       | z        | 150.5 | 87.0  | 16   | 17    |

SELECT \* FROM vzťah

| id | id_on | id_ona | od                  | do                  |
|----|-------|--------|---------------------|---------------------|
| 1  | 1     | 2      | 1937-06-01 00:00:00 | 1967-05-11 00:00:00 |
| 2  | 3     | 2      | 1967-05-12 00:00:00 | 1988-07-22 00:00:00 |
| 3  | 3     | 4      | 1938-12-02 00:00:00 | 1965-03-11 00:00:00 |
| 4  | 5     | 6      | 1953-11-11 00:00:00 | NULL                |
| 5  | 7     | 8      | 1970-07-22 00:00:00 | 1975-09-01 00:00:00 |
| 6  | 11    | 12     | 1980-03-04 00:00:00 | 2000-12-31 00:00:00 |
| 7  | 16    | 15     | 1997-07-31 00:00:00 | 2002-12-04 00:00:00 |

# Prečo je normalizácia dôležitá?

## (Anomálie v praxi)

- Predstavte si, že by normalizácia neexistovala a vy by ste informácie o vzťahoch uložili priamo do tabuľky Osoba (napr. stĺpce partner\_meno, od, do). V praxi by to spôsobilo tri fatálne problémy, ktoré akademicky nazývame anomálie:
- **Redundancia (Nadbytočnosť):** Ak by boli Adam a Eva vo vzťahu, museli by ste dátum začiatku ich vzťahu (1937-06-01) zapísať dvakrát – raz k riadku Adama a raz k riadku Evy. Zbytočne tak plytváte diskovým priestorom.
- **Anomália pri aktualizácii (Update Anomaly):** Ak by sa zistilo, že dátum vzťahu je nesprávny, museli by ste ho prepísať na dvoch miestach (u oboch partnerov). Ak na jedného zabudnete, databáza sa dostane do **nekonzistentného stavu** (dáta si budú protirečiť).
- **Anomália pri mazaní (Delete Anomaly):** Ak by ste chceli zmazať informáciu o tom, že vzťah skončil, a zmazali by ste celý riadok, mohli by ste omylom vymazať aj samotnú identitu osoby (jej meno, výšku či váhu), čo je neprípustné.

# Normalizácia v praxi

- Vy ste tento problém vyriešili profesionálne pomocou **dekompozície relácií** do dvoch samostatných entitných množín:
  1. **Tabuľka Osoba** drží striktne a výhradne atribúty konkrétneho človeka (kto to je, ako sa volá, kedy sa narodil, akú má výšku a váhu).
  2. **Tabuľka Vzťah** slúži ako tzv. **prepojovacia (spojovacia) tabuľka**. Neobsahuje mená ani fyzické vlastnosti. Obsahuje iba **cudzíe kľúče** (id\_on, id\_ona), ktoré odkazujú na primárne kľúče z tabuľky Osoba, a dopĺňa ich o atribúty samotného vzťahu (od, do).
- Vďaka tomu:
  - Ak sa zmení váha Jozefa Urbana, opravíte ju na **jednom jedinom mieste** v tabuľke Osoba. Táto zmena sa okamžite správne prejaví vo všetkých jeho vzťahoch, pretože tabuľka Vzťah iba odkazuje na jeho id (5).
  - Dátum začiatku vzťahu je zapísaný **iba raz** pre danú dvojicu.

# Normálne formy

- Normalizácia prebieha v po sebe idúcich krokoch, ktoré nazývame **Normálne formy (NF)**. Aby bola databáza považovaná za zdravú, musí v komerčnej praxi spĺňať prvé tri:
- **1. Normálna forma (1NF):** Všetky atribúty musia byť atomárne (nedeliteľné). V bunke nesmie byť zoznam hodnôt (napr. v stĺpci meno nemôže byť napísané 'Adam a Eva'). Každý riadok musí byť unikátny (musí existovať primárny kľúč).
- **2. Normálna forma (2NF):** Splnená 1NF + všetky neklúčové stĺpce musia byť plne závislé od celého primárneho kľúča (rieši sa najmä pri zložených kľúčoch).
- **3. Normálna forma (3NF):** Splnená 2NF + medzi neklúčovými stĺpcami nesmie existovať tranzitívna závislosť. To znamená, že žiadny neklúčový stĺpec nesmie závisieť od iného neklúčového stĺpca (napríklad, ak by ste mali v tabuľke stĺpce PSC a Mesto, porušujete 3NF, pretože názov mesta viete odvodiť z PSC a tieto dáta patria do samostatnej číselníkovej tabuľky).
- **Heat-Boyce-Coddova normálna forma (HBCNF):** Sprísňuje tretiu normálnu formu tým, že vyžaduje, aby každý stĺpec, ktorý určuje hodnotu iného stĺpca, bol striktne kandidátskym kľúčom.

# Normálne formy

- **4. Normálna forma (4NF):** Eliminuje takzvané viachodnotové závislosti, čím zabraňuje tomu, aby sa v jednej tabuľke museli zbytočne kombinovať dva nezávislé vzťahy typu 1:N pre tú istú entitu.
- **Essential Tuple Normálna Forma (ETNF):** Redukuje nadbytočnosť dát tým, že striktne oddeľuje funkčné závislosti od závislostí spojenia, čím zabezpečuje, že relácia nemôže byť dekomponovaná bez straty dôležitých štrukturálnych väzieb.
- **5. Normálna forma (5NF):** Zabezpečuje, že zložité cyklické vzťahy medzi viacerými entitami je možné bezstratovo rozdeliť do samostatných tabuliek a následne ich bez vzniku falošných riadkov spojiť pomocou operácie JOIN.
- **Doménovo-klúčová normálna forma (DKNF):** Predstavuje teoretické maximum normalizácie, kedy sú všetky biznis pravidlá a integritné obmedzenia priamym dôsledkom definície kľúčov a povolených hodnôt stĺpcov, čo absolútne vylučuje akékoľvek anomálie.
- **6. Normálna forma (6NF):** Predstavuje dekompozíciu dotiahnutú do absolútneho extrému, kedy každá tabuľka obsahuje iba primárny kľúč a maximálne jeden ďalší neklúčový stĺpec, čo nachádza uplatnenie najmä v temporálnych databázach sledujúcich históriu zmien v čase.

# Normálne formy

- Normalizácia je inherentne **inkluzívny a kumulatívny proces**.  
Relácia **nemôže** dosiahnuť vyššiu normálnu formu (napr. 4NF) bez toho, aby predtým stopercentne spĺňala podmienky **všetkých predchádzajúcich úrovní** (1NF → 2NF → 3NF → HBCNF → 4NF...).
- Vyššie normálne formy stavajú na základoch tých nižších a ďalej ich sprísňujú.

| Normálna forma | Hlavný cieľ (Čo odstraňuje / Čo vynucuje)                              | Typická anomália / Riešený problém                                           |
|----------------|------------------------------------------------------------------------|------------------------------------------------------------------------------|
| <b>1NF</b>     | Vynucuje atomicitu hodnôt atribútov a existenciu primárneho kľúča.     | Odstraňuje zoznamy a duplicitné riadky.                                      |
| <b>2NF</b>     | Odstraňuje čiastočnú závislosť neklúčových stĺpcov na zloženom kľúči.  | Rieši redundantné opakovanie dát pri zložených kľúčoch.                      |
| <b>3NF</b>     | Odstraňuje tranzitívnu závislosť medzi neklúčovými stĺpcami.           | Zamedzuje, aby stĺpec A určoval hodnotu stĺpca B mimo kľúča.                 |
| <b>HBCNF</b>   | Zaisťuje, že každý determinant (určujúci stĺpec) musí byť superkľúčom. | Rieši prekryvajúce sa zložené kandidátske kľúče.                             |
| <b>4NF</b>     | Odstraňuje viachodnotové závislosti (MVD).                             | Zamedzuje nutnosti držať v tabuľke kombinácie dvoch nezávislých vzťahov 1:N. |
| <b>5NF</b>     | Odstraňuje anomálie pri viacnásobnom cyklickom spájaní relácií.        | Zaručuje 100 % bezstratovú dekompozíciu a opätovný JOIN.                     |
| <b>DKNF</b>    | Zjednocuje všetky pravidlá pod definície domén a kľúčov.               | Absolútna eliminácia akýchkoľvek logických anomálií modifikácie.             |

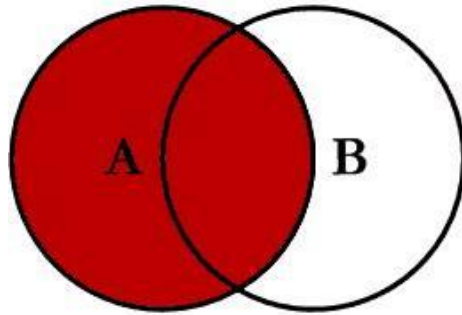
# Princíp normalizácie a prepojenia relácií (JOIN)

- **Prečo nedržíme všetky dáta v jednej tabuľke?** Koncept normalizácie dát (eliminácia redundancie, anomálií pri vkladaní/aktualizácii a úspora diskového priestoru).
- **Naša schéma:** Informácie o entitách (osoba) sú oddelené od informácií o ich interakciách (vztah).
- **Mechanizmus prepojenia:** Využitie **primárnych kľúčov** (osoba.id) a **cudzích kľúčov** (vztah.id\_on, vztah.id\_ona).
- **Definícia JOIN:** Operácia Princíp normalizácie a prepojenia relácií (JOIN) a relačnej algebry, ktorá na základe spojovacieho predikátu (podmienky) zlúči atribúty dvoch relácií do novej výslednej množiny.

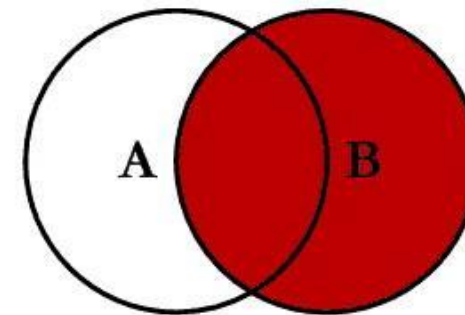
# Taxonómia operácií JOIN

- Podľa exekučnej logiky a správania k nezhodným riadkam delíme spájanie na:
  - **CROSS JOIN (kartézsky súčin):** Kombinácia každého riadku s každým ( $M \times N$  riadkov).
  - **INNER JOIN (vnútorné spojenie):** Striktný prienik množín. Vracia len n-tice, ktoré majú zhodu v oboch reláciách.
  - **LEFT / RIGHT / FULL OUTER JOIN (vonkajšie spojenie):** Zachováva kompletnú riadkovú množinu jednej (alebo oboch) tabuliek a chýbajúce protipóly dopĺňa nedefinovanou hodnotou **NULL**.
  - **SELF JOIN (samo-spojenie):** Unárna operácia, kedy tabuľka vstupuje do dopytu dvakrát v dvoch rôznych rolách.
  - **NATURAL JOIN (prirodzené spojenie):** Ide o implicitné spájanie, pri ktorom databázový engine **automaticky vyhľadá stĺpce s identickým názvom** v oboch tabuľkách a spojí ich. V dopyte sa nepíše klauzula **ON**.

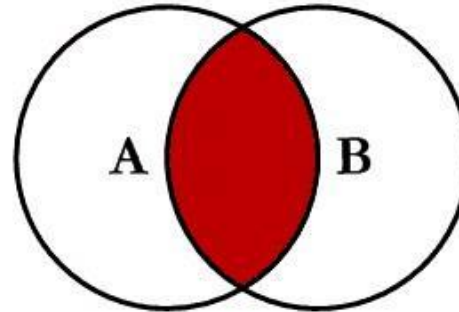
# SQL JOINS



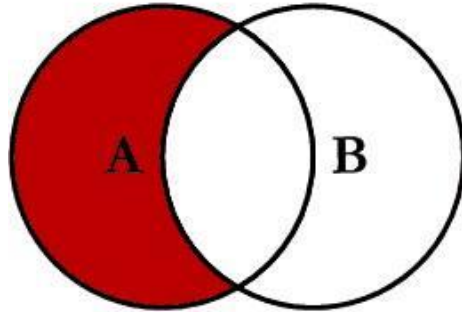
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



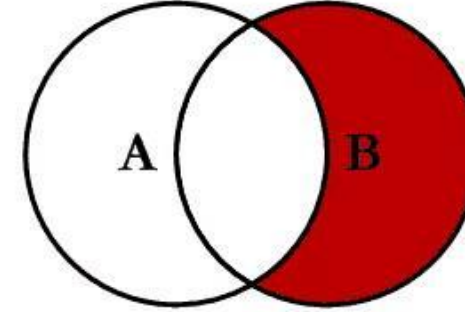
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



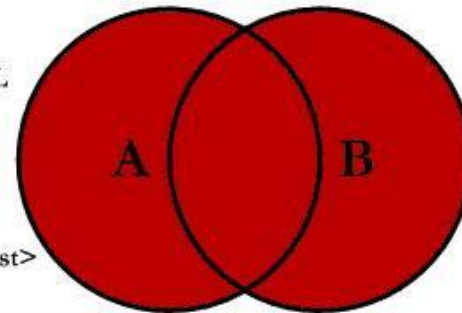
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



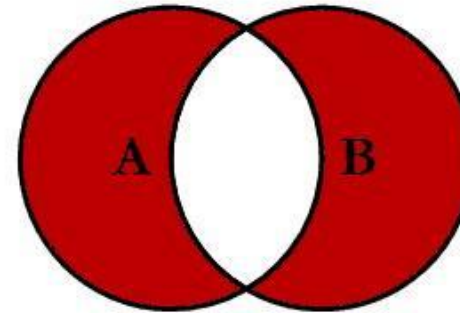
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```

# Ďalšie spôsoby spojenia tabuliek

- **STRAIGHT\_JOIN:** ako INNER JOIN, ale vynúti poradie tabuliek
- **SEMI JOIN:** vracia riadky z A, ak existuje zhoda v B (bez duplikácie) - MySQL nepodporuje, ale napr. MS SQL Server áno
- **ANTI JOIN:** vracia riadky z A, ktoré nemajú zhodu v B (napr. NOT EXISTS) - MySQL nepodporuje, ale napr. MS SQL Server áno
- **LOOP | HASH | MERGE | REMOTE | REDUCE | REPLICATE | REDISTRIBUTE [(columns count)] JOIN:** MySQL nepodporuje, ale napr. MS SQL Server áno
- **Parallel join:** join rozdelený medzi viacero vlákien - MySQL nepodporuje, ale napr. MS SQL Server áno
- **Materialized join:** jedna strana joinu je v dočasnej tabuľke
- ...

# CROSS JOIN

- Tento typ spojenia vytvorí kombináciu **každého riadku z prvej tabuľky s každým riadkom z druhej tabuľky**.
- **SELECT \* FROM** osoba, vzťah; -- 23 x 7 = 161 riadkov

| id | meno | priezvisko | rodne_priezvisko | dat_nar    | dat_smrti  | pohlavie | vyska | vaha | otec | matka | id | id_on | id_ona | od                  | do                  |
|----|------|------------|------------------|------------|------------|----------|-------|------|------|-------|----|-------|--------|---------------------|---------------------|
| 1  | Adam | Prvý       | NULL             | 1918-05-11 | 1968-10-01 | m        | 180.0 | 80.0 | NULL | NULL  | 7  | 16    | 15     | 1997-07-31 00:00:00 | 2002-12-04 00:00:00 |
| 1  | Adam | Prvý       | NULL             | 1918-05-11 | 1968-10-01 | m        | 180.0 | 80.0 | NULL | NULL  | 6  | 11    | 12     | 1980-03-04 00:00:00 | 2000-12-31 00:00:00 |
| 1  | Adam | Prvý       | NULL             | 1918-05-11 | 1968-10-01 | m        | 180.0 | 80.0 | NULL | NULL  | 5  | 7     | 8      | 1970-07-22 00:00:00 | 1975-09-01 00:00:00 |
| 1  | Adam | Prvý       | NULL             | 1918-05-11 | 1968-10-01 | m        | 180.0 | 80.0 | NULL | NULL  | 4  | 5     | 6      | 1953-11-11 00:00:00 | NULL                |
| 1  | Adam | Prvý       | NULL             | 1918-05-11 | 1968-10-01 | m        | 180.0 | 80.0 | NULL | NULL  | 3  | 3     | 4      | 1938-12-02 00:00:00 | 1965-03-11 00:00:00 |
| 1  | Adam | Prvý       | NULL             | 1918-05-11 | 1968-10-01 | m        | 180.0 | 80.0 | NULL | NULL  | 2  | 3     | 2      | 1967-05-12 00:00:00 | 1988-07-22 00:00:00 |
| 1  | Adam | Prvý       | NULL             | 1918-05-11 | 1968-10-01 | m        | 180.0 | 80.0 | NULL | NULL  | 1  | 1     | 2      | 1937-06-01 00:00:00 | 1967-05-11 00:00:00 |
| 2  | Eva  | Prvá       | Druhá            | 1919-01-09 | 1988-07-22 | z        | 160.0 | 60.0 | NULL | NULL  | 7  | 16    | 15     | 1997-07-31 00:00:00 | 2002-12-04 00:00:00 |
| 2  | Eva  | Prvá       | Druhá            | 1919-01-09 | 1988-07-22 | z        | 160.0 | 60.0 | NULL | NULL  | 6  | 11    | 12     | 1980-03-04 00:00:00 | 2000-12-31 00:00:00 |
| 2  | Eva  | Prvá       | Druhá            | 1919-01-09 | 1988-07-22 | z        | 160.0 | 60.0 | NULL | NULL  | 5  | 7     | 8      | 1970-07-22 00:00:00 | 1975-09-01 00:00:00 |
| 2  | Eva  | Prvá       | Druhá            | 1919-01-09 | 1988-07-22 | z        | 160.0 | 60.0 | NULL | NULL  | 4  | 5     | 6      | 1953-11-11 00:00:00 | NULL                |
| 2  | Eva  | Prvá       | Druhá            | 1919-01-09 | 1988-07-22 | z        | 160.0 | 60.0 | NULL | NULL  | 3  | 3     | 4      | 1938-12-02 00:00:00 | 1965-03-11 00:00:00 |
| 2  | Eva  | Prvá       | Druhá            | 1919-01-09 | 1988-07-22 | z        | 160.0 | 60.0 | NULL | NULL  | 2  | 3     | 2      | 1967-05-12 00:00:00 | 1988-07-22 00:00:00 |
| 2  | Eva  | Prvá       | Druhá            | 1919-01-09 | 1988-07-22 | z        | 160.0 | 60.0 | NULL | NULL  | 1  | 1     | 2      | 1937-06-01 00:00:00 | 1967-05-11 00:00:00 |
| 3  | Zoly | Múdry      | NULL             | 1918-04-07 | 1990-09-23 | m        | 175.5 | 75.0 | NULL | NULL  | 7  | 16    | 15     | 1997-07-31 00:00:00 | 2002-12-04 00:00:00 |
| 3  | Zoly | Múdry      | NULL             | 1918-04-07 | 1990-09-23 | m        | 175.5 | 75.0 | NULL | NULL  | 6  | 11    | 12     | 1980-03-04 00:00:00 | 2000-12-31 00:00:00 |

# CROSS JOIN

- `SELECT * FROM osoba AS o, vzťah v WHERE v.id_on = o.id;`  
-- 7 riadkov
- `<=>`
- `SELECT * FROM osoba AS o CROSS JOIN vzťah v WHERE v.id_on = o.id;`  
-- 7 riadkov

| id | meno     | priezvisko | rodne_priezvisko | dat_nar    | dat_smrti  | pohlavie | vyska | vaha  | otec | matka | id | id_on | id_ona | od                  | do                  |
|----|----------|------------|------------------|------------|------------|----------|-------|-------|------|-------|----|-------|--------|---------------------|---------------------|
| 1  | Adam     | Prvý       | NULL             | 1918-05-11 | 1968-10-01 | m        | 180.0 | 80.0  | NULL | NULL  | 1  | 1     | 2      | 1937-06-01 00:00:00 | 1967-05-11 00:00:00 |
| 3  | Zoly     | Múdry      | NULL             | 1918-04-07 | 1990-09-23 | m        | 175.5 | 75.0  | NULL | NULL  | 2  | 3     | 2      | 1967-05-12 00:00:00 | 1988-07-22 00:00:00 |
| 3  | Zoly     | Múdry      | NULL             | 1918-04-07 | 1990-09-23 | m        | 175.5 | 75.0  | NULL | NULL  | 3  | 3     | 4      | 1938-12-02 00:00:00 | 1965-03-11 00:00:00 |
| 5  | Jozef    | Urban      | NULL             | 1922-10-19 | NULL       | m        | 199.5 | NULL  | NULL | NULL  | 4  | 5     | 6      | 1953-11-11 00:00:00 | NULL                |
| 7  | Patrik   | Novák      | Novák            | 1945-06-19 | NULL       | m        | 182.5 | 89.5  | 1    | 2     | 5  | 7     | 8      | 1970-07-22 00:00:00 | 1975-09-01 00:00:00 |
| 11 | Peter    | Horváth    | Horváth          | 1959-07-02 | 2000-12-31 | m        | 193.0 | 110.5 | NULL | NULL  | 6  | 11    | 12     | 1980-03-04 00:00:00 | 2000-12-31 00:00:00 |
| 16 | Vladimír | Silný      | Silný            | 1974-08-01 | 2002-12-04 | m        | 175.5 | 73.0  | NULL | NULL  | 7  | 16    | 15     | 1997-07-31 00:00:00 | 2002-12-04 00:00:00 |

# CROSS JOIN

- Databáza najprv v prvej fáze vygeneruje v pamäti RAM kompletný **CROSS JOIN (Kartézsky súčin)** oboch tabuliek. Nad vašimi dátami tak vytvorí obrovskú dočasnú tabuľku so **161 riadkami** ( $23 \text{ osôb} \times 7 \text{ vzťahov}$ ).
- **Až v druhej fáze** začne databáza sekvenčne prechádzať týchto 161 riadkov a filtrovať podmienkou **WHERE v.id\_on = o.id**, kým jej nezostane výsledných 7 správnych riadkov.
- **Dôsledok pre výkon:** Pri malých tabuľkách si to nevšimnete, ale ak by tabuľka osoba mala **10 000 riadkov** a vzťah **10 000 riadkov**, databáza najprv v pamäti vytvorí tabuľku so **100 miliónmi riadkov**, čo môže server okamžite zahltiť, spomaliť alebo úplne zamraziť systém.

# Query Optimizer

- Moderné systémy riadenia báz dát (RDBMS) sa spoliehajú na sofistikovaný modul **Query Optimizer (optimalizátor dopytov)**, ktorý na základe štatistík diskových úložísk a dostupnosti indexových štruktúr zostavuje optimálny exekučný plán (**Execution Plan**).
- Pri **explicitnej** deklarácii pomocou syntaxe **INNER JOIN ... ON** dáva vývojár **jednoznačnú sémantickú inštrukciu o spôsobe prepájania tabuliek**. Optimizátor dokáže okamžite aplikovať pokročilé vyhľadávacie algoritmy (napr. *Hash Join* alebo *Merge Join*) a využiť B-stromové indexy na cudzích kľúčoch, čím minimalizuje diskové operácie (I/O).
- Pri použití **implicitného** zápisu (čiarka v klauzule **FROM**) je optimalizátor nútený vykonať **dodatočnú sémantickú analýzu**. Hoci sa moderné RDBMS snažia tento starý zápis transformovať na explicitný JOIN za behu, táto heuristická optimalizácia má svoje limity. Pri nelineárne rastúcej komplexnosti dopytu (ak spájame viacero relácií súčasne s viacerými logickými operátormi) dochádza k **destrukcii exekučného plánu**. Výsledkom je, že databázový engine často zlyhá pri voľbe správneho indexu, čo vedie k sekvenčnému prechádzaniu fyzického úložiska (**Full Table Scan**) a degradácii celkového výkonu systému.

# INNER JOIN

- Spojenie nastane, ak podmienka v klauzule **ON** vráti hodnotu **TRUE**.
- Ak osoba nie je zapísaná v žiadnom vzťahu, vo výsledku sa vôbec neobjaví.

- **SELECT**

```
o.id AS osoba_id,  
o.meno,  
o.priezvisko,  
v.id AS vzťah_id,  
v.od,  
v.do
```

```
FROM osoba AS o
```

```
INNER JOIN vzťah AS v ON v.id_on = o.id;
```

```
-- 7 riadkov
```

- **SELECT**

```
o.id osoba_id,  
meno,  
priezvisko,  
v.id vzťah_id,  
od,  
do
```

```
FROM osoba o
```

```
JOIN vzťah v ON o.id = v.id_on;
```

```
-- 7 riadkov
```

| osoba_id | meno     | priezvisko | vzťah_id | od                  | do                  |
|----------|----------|------------|----------|---------------------|---------------------|
| 1        | Adam     | Prvý       | 1        | 1937-06-01 00:00:00 | 1967-05-11 00:00:00 |
| 3        | Zoly     | Múdry      | 2        | 1967-05-12 00:00:00 | 1988-07-22 00:00:00 |
| 3        | Zoly     | Múdry      | 3        | 1938-12-02 00:00:00 | 1965-03-11 00:00:00 |
| 5        | Jozef    | Urban      | 4        | 1953-11-11 00:00:00 | NULL                |
| 7        | Patrik   | Novák      | 5        | 1970-07-22 00:00:00 | 1975-09-01 00:00:00 |
| 11       | Peter    | Horváth    | 6        | 1980-03-04 00:00:00 | 2000-12-31 00:00:00 |
| 16       | Vladimír | Silný      | 7        | 1997-07-31 00:00:00 | 2002-12-04 00:00:00 |

# LEFT OUTER JOIN

- Garantuje, že z **ľavej tabuľky** (uvedenej pred slovom JOIN) prejdú do výstupu **všetky riadky**.
- Ak pre riadok z ľavej tabuľky existuje zhoda v pravej, dáta sa prepoja.
- Ak zhoda neexistuje, atribúty pravej tabuľky sa vo výslednej relácii vyplnia hodnotou **NULL**.

• -- 24 riadkov

- **SELECT**

o.id **AS** osoba\_id,  
o.meno,  
o.priezvisko,  
v.id **AS** vzťah\_id,  
v.od,  
v.do

**FROM** osoba **AS** o

**LEFT OUTER JOIN** vzťah **AS** v **ON** o.id = v.id\_on;

-- LEFT JOIN vzťah **AS** v **ON** o.id = v.id\_on;

| osoba_id | meno     | priezvisko | vztah_id | od                  | do                  |
|----------|----------|------------|----------|---------------------|---------------------|
| 1        | Adam     | Prvý       | 1        | 1937-06-01 00:00:00 | 1967-05-11 00:00:00 |
| 2        | Eva      | Prvá       | NULL     | NULL                | NULL                |
| 3        | Zoly     | Múdry      | 3        | 1938-12-02 00:00:00 | 1965-03-11 00:00:00 |
| 3        | Zoly     | Múdry      | 2        | 1967-05-12 00:00:00 | 1988-07-22 00:00:00 |
| 4        | Nasťa    | Kováčová   | NULL     | NULL                | NULL                |
| 5        | Jozef    | Urban      | 4        | 1953-11-11 00:00:00 | NULL                |
| 6        | Mária    | Urbanová   | NULL     | NULL                | NULL                |
| 7        | Patrik   | Novák      | 5        | 1970-07-22 00:00:00 | 1975-09-01 00:00:00 |
| 8        | Patricia | Nováková   | NULL     | NULL                | NULL                |
| 9        | Michal   | Kováč      | NULL     | NULL                | NULL                |
| 10       | Roman    | Kováč      | NULL     | NULL                | NULL                |
| 11       | Peter    | Horváth    | 6        | 1980-03-04 00:00:00 | 2000-12-31 00:00:00 |
| 12       | Lucia    | Horváthová | NULL     | NULL                | NULL                |
| 13       | Urban    | Urban      | NULL     | NULL                | NULL                |
| 14       | Dáša     | Nováková   | NULL     | NULL                | NULL                |
| 15       | Viera    | Lucia ná   | NULL     | NULL                | NULL                |
| 16       | Vladimír | Silný      | 7        | 1997-07-31 00:00:00 | 2002-12-04 00:00:00 |
| 17       | Milena   | Slabá      | NULL     | NULL                | NULL                |
| 18       | Ján      | Horváth    | NULL     | NULL                | NULL                |
| 19       | Zuzana   | Silná      | NULL     | NULL                | NULL                |
| 20       | Zuzana   | Slabá      | NULL     | NULL                | NULL                |
| 21       | Zuzana   | Pravá      | NULL     | NULL                | NULL                |
| 22       | Zuzana   | Ľavá       | NULL     | NULL                | NULL                |
| 23       | Zuzana   | Stredná    | NULL     | NULL                | NULL                |

# POZOR na podmienky

- **Zadanie:** Chceme zistiť informácie o vzťahu **len pre Patrika (id 7)**, ale chceme vidieť aj zoznam ostatných ľudí.
- **NESPRÁVNE**
- **SELECT** o.id, o.meno, o.priezvisko, v.\*  
**FROM** osoba **AS** o  
    **LEFT JOIN** vzťah **AS** v **ON** o.id = v.id\_on  
**WHERE** o.id = 7;

| id | meno   | priezvisko | id | id_on | id_ona | od                  | do                  |
|----|--------|------------|----|-------|--------|---------------------|---------------------|
| 7  | Patrik | Novák      | 5  | 7     | 8      | 1970-07-22 00:00:00 | 1975-09-01 00:00:00 |

# POZOR na podmienky

- **Zadanie:** Chceme zistiť informácie o vzťahu **len pre Patrika (id 7)**, ale chceme vidieť aj zoznam ostatných ľudí.

- **SPRÁVNE**

- **SELECT** o.id, o.meno, o.priezvisko, v.\*  
**FROM** osoba **AS** o  
**LEFT JOIN** vzťah **AS** v  
**ON** o.id = v.id\_on **AND** o.id = 7;

| id | meno     | priezvisko | id   | id_on | id_ona | od                  | do                  |
|----|----------|------------|------|-------|--------|---------------------|---------------------|
| 1  | Adam     | Prvý       | NULL | NULL  | NULL   | NULL                | NULL                |
| 2  | Eva      | Prvá       | NULL | NULL  | NULL   | NULL                | NULL                |
| 3  | Zoly     | Múdry      | NULL | NULL  | NULL   | NULL                | NULL                |
| 4  | Nasťa    | Kováčová   | NULL | NULL  | NULL   | NULL                | NULL                |
| 5  | Jozef    | Urban      | NULL | NULL  | NULL   | NULL                | NULL                |
| 6  | Mária    | Urbanová   | NULL | NULL  | NULL   | NULL                | NULL                |
| 7  | Patrik   | Novák      | 5    | 7     | 8      | 1970-07-22 00:00:00 | 1975-09-01 00:00:00 |
| 8  | Patrícia | Nováková   | NULL | NULL  | NULL   | NULL                | NULL                |
| 9  | Michal   | Kováč      | NULL | NULL  | NULL   | NULL                | NULL                |
| 10 | Roman    | Kováč      | NULL | NULL  | NULL   | NULL                | NULL                |
| 11 | Peter    | Horváth    | NULL | NULL  | NULL   | NULL                | NULL                |
| 12 | Lucia    | Horváthová | NULL | NULL  | NULL   | NULL                | NULL                |
| 13 | Urban    | Urban      | NULL | NULL  | NULL   | NULL                | NULL                |
| 14 | Dáša     | Nováková   | NULL | NULL  | NULL   | NULL                | NULL                |
| 15 | Viera    | Silná      | NULL | NULL  | NULL   | NULL                | NULL                |
| 16 | Vladimír | Silný      | NULL | NULL  | NULL   | NULL                | NULL                |
| 17 | Milena   | Slabá      | NULL | NULL  | NULL   | NULL                | NULL                |
| 18 | Ján      | Horváth    | NULL | NULL  | NULL   | NULL                | NULL                |
| 19 | Zuzana   | Silná      | NULL | NULL  | NULL   | NULL                | NULL                |
| 20 | Zuzana   | Slabá      | NULL | NULL  | NULL   | NULL                | NULL                |
| 21 | Zuzana   | Pravá      | NULL | NULL  | NULL   | NULL                | NULL                |
| 22 | Zuzana   | Ľavá       | NULL | NULL  | NULL   | NULL                | NULL                |
| 23 | Zuzana   | Stredná    | NULL | NULL  | NULL   | NULL                | NULL                |

# RIGHT OUTER JOIN

- Zrkadlový protipól **LEFT OUTER JOIN**.
- Plne zachováva riadkovú množinu pravej relácie.
- Ak v ľavej tabuľke neexistuje entita spĺňajúca podmienku, atribúty ľavej strany sú reprezentované ako **NULL**.

- -- 7 riadkov

- **SELECT**

- o.id **AS** osoba\_id,
  - o.meno,
  - o.priezvisko,
  - v.id **AS** vzťah\_id,
  - v.od,
  - v.do

- FROM** osoba **AS** o

- RIGHT OUTER JOIN** vzťah **AS** v **ON** o.id = v.id\_on;

- RIGHT JOIN Vzťah AS v ON o.id = v.id\_on;

| osoba_id | meno     | priezvisko | vztah_id | od                  | do                  |
|----------|----------|------------|----------|---------------------|---------------------|
| 1        | Adam     | Prvý       | 1        | 1937-06-01 00:00:00 | 1967-05-11 00:00:00 |
| 3        | Zoly     | Múdry      | 2        | 1967-05-12 00:00:00 | 1988-07-22 00:00:00 |
| 3        | Zoly     | Múdry      | 3        | 1938-12-02 00:00:00 | 1965-03-11 00:00:00 |
| 5        | Jozef    | Urban      | 4        | 1953-11-11 00:00:00 | NULL                |
| 7        | Patrik   | Novák      | 5        | 1970-07-22 00:00:00 | 1975-09-01 00:00:00 |
| 11       | Peter    | Horváth    | 6        | 1980-03-04 00:00:00 | 2000-12-31 00:00:00 |
| 16       | Vladimír | Silný      | 7        | 1997-07-31 00:00:00 | 2002-12-04 00:00:00 |

# Prepis RIGHT OUTER JOIN na LEFT OUTER JOIN

- **Symetria:** Operácie **LEFT OUTER JOIN** a **RIGHT OUTER JOIN** sú matematicky identické, líšia sa iba sémantickým určením toho, ktorá relácia je považovaná za dominantnú (zachovanú) a ktorá za komplementárnu (doplňanú o **NULL**).
- **Pravidlo transformácie:** Akúkoľvek exekučnú šablónu typu **T1 RIGHT OUTER JOIN T2** je možné transformovať na plne ekvivalentný zápis **T2 LEFT OUTER JOIN T1**. Spojovací predikát v klauzule **ON** zostáva navlas rovnaký.
- **Odporúčanie:** V profesionálnom vývoji sa používanie **RIGHT OUTER JOIN** **takmer vôbec neodporúča**. Miešanie **LEFT** a **RIGHT** spojení v jednom komplexnom dopyte znižuje čitateľnosť kódu. Štandardom je držať sa schémy čítania zľava doprava a vrstviť dopyt výhradne cez **LEFT OUTER JOIN**.

# Prepis RIGHT OUTER JOIN na LEFT OUTER JOIN

-- 7 riadkov

SELECT

o.id AS osoba\_id,  
o.meno,  
o.priezvisko,  
v.id AS vzťah\_id,  
v.od,  
v.do

FROM osoba AS o

RIGHT OUTER JOIN vzťah AS v ON o.id = v.id\_on;

-- RIGHT JOIN Vzťah AS v ON o.id = v.id\_on;

-- <=>

SELECT

o.id AS osoba\_id,  
o.meno,  
o.priezvisko,  
v.id AS vzťah\_id,  
v.od,  
v.do

FROM vzťah AS o

LEFT OUTER JOIN osoba AS o ON o.id = v.id\_on;

-- LEFT OUTER JOIN osoba AS o ON o.id = v.id\_on;

| osoba_id | meno     | priezvisko | vzťah_id | od                  | do                  |
|----------|----------|------------|----------|---------------------|---------------------|
| 1        | Adam     | Prvý       | 1        | 1937-06-01 00:00:00 | 1967-05-11 00:00:00 |
| 3        | Zoly     | Múdry      | 2        | 1967-05-12 00:00:00 | 1988-07-22 00:00:00 |
| 3        | Zoly     | Múdry      | 3        | 1938-12-02 00:00:00 | 1965-03-11 00:00:00 |
| 5        | Jozef    | Urban      | 4        | 1953-11-11 00:00:00 | NULL                |
| 7        | Patrik   | Novák      | 5        | 1970-07-22 00:00:00 | 1975-09-01 00:00:00 |
| 11       | Peter    | Horváth    | 6        | 1980-03-04 00:00:00 | 2000-12-31 00:00:00 |
| 16       | Vladimír | Silný      | 7        | 1997-07-31 00:00:00 | 2002-12-04 00:00:00 |

# FULL OUTER JOIN

- Zjednotenie množín. Kombinuje logiku **LEFT** aj **RIGHT OUTER JOIN**.
- Ponecháva všetky riadky z oboch tabuliek a prázdne väzby z oboch strán dopĺňa hodnotami **NULL**.

```
• SELECT  
    o.meno,  
    o.priezvisko,  
    v.id AS id_vztahu,  
    v.od  
FROM osoba AS o  
    FULL OUTER JOIN vztah AS v  
        ON o.id = v.id_on;  
-- 24 riadkov
```

| osoba_id | meno     | priezvisko | vztah_id | od                  |
|----------|----------|------------|----------|---------------------|
| 1        | Adam     | Prvý       | 1        | 1937-06-01 00:00:00 |
| 2        | Eva      | Prvá       | NULL     | NULL                |
| 3        | Zoly     | Múdry      | 3        | 1938-12-02 00:00:00 |
| 3        | Zoly     | Múdry      | 2        | 1967-05-12 00:00:00 |
| 4        | Nastá    | Kováčová   | NULL     | NULL                |
| 5        | Jozef    | Urban      | 4        | 1953-11-11 00:00:00 |
| 6        | Mária    | Urbanová   | NULL     | NULL                |
| 7        | Patrik   | Novák      | 5        | 1970-07-22 00:00:00 |
| 8        | Patricia | Nováková   | NULL     | NULL                |
| 9        | Michal   | Kováč      | NULL     | NULL                |
| 10       | Roman    | Kováč      | NULL     | NULL                |
| 11       | Peter    | Horváth    | 6        | 1980-03-04 00:00:00 |
| 12       | Lucia    | Horváthová | NULL     | NULL                |
| 13       | Urban    | Urban      | NULL     | NULL                |
| 14       | DÁŠa     | Nováková   | NULL     | NULL                |
| 15       | Viera    | Silná      | NULL     | NULL                |
| 16       | Vladimír | Silný      | 7        | 1997-07-31 00:00:00 |
| 17       | Milena   | Slabá      | NULL     | NULL                |
| 18       | Ján      | Horváth    | NULL     | NULL                |
| 19       | Zuzana   | Silná      | NULL     | NULL                |
| 20       | Zuzana   | Slabá      | NULL     | NULL                |
| 21       | Zuzana   | Pravá      | NULL     | NULL                |
| 22       | Zuzana   | Ľavá       | NULL     | NULL                |
| 23       | Zuzana   | Stredná    | NULL     | NULL                |

# Emulácia FULL OUTER JOIN

- **MySQL nepodporuje natívny FULL OUTER JOIN.** Ak sa pokúsite použiť toto kľúčové slovo, SQL parser vyhodí chybu syntaxe.
- Túto funkciu však môžete jednoducho **nasimulovať spojením LEFT OUTER JOIN a RIGHT OUTER JOIN** pomocou operátora **UNION**.

-- 1. Krok: Načítame kompletnú ľavú stranu (všetky osoby, aj tie bez vzťahu)

```
SELECT o.id AS osoba_id, o.meno, o.priezvisko, v.id AS vzťah_id, v.od  
FROM osoba AS o
```

```
    LEFT OUTER JOIN vzťah AS v ON v.id_on = o.id
```

```
UNION -- Operátor zjednotenia (automaticky odstráni duplicitné n-tice)
```

-- 2. Krok: Načítame kompletnú pravú stranu (všetky vzťahy, aj tie bez osôb)

```
SELECT o.id AS osoba_id, o.meno, o.priezvisko, v.id AS vzťah_id, v.od  
FROM osoba AS o
```

```
    RIGHT OUTER JOIN vzťah AS v ON v.id_on = o.id;
```

# SELF JOIN (Samo-spojenie / Unárne spájanie)

- Ide o spájanie **tabuľky so sebou samou**. V relačnej algebre ide o unárnu operáciu. Používa sa vtedy, keď tabuľka obsahuje rekurzívny cudzí kľúč (atribút, ktorý odkazuje na primárny kľúč v tej istej tabuľke).
- Napr. tabuľka osoba obsahuje stĺpce otec a matka, ktoré referencujú osoba(id).

```
• SELECT
    dieta.meno AS meno_dietata,
    dieta.priezvisko AS priezvisko_dietata,
    mama.meno AS meno_matky,
    mama.priezvisko AS priezvisko_matky
FROM Osoba AS dieta
JOIN Osoba AS mama
    ON dieta.matka = mama.id;
-- 14 riadkov
```

| meno_dietata | priezvisko_dietata | meno_matky | priezvisko_matky |
|--------------|--------------------|------------|------------------|
| Mária        | Urbanová           | Eva        | Prvá             |
| Patrik       | Novák              | Eva        | Prvá             |
| Michal       | Kováč              | Eva        | Prvá             |
| Roman        | Kováč              | Nasťa      | Kováčová         |
| Lucia        | Horváthová         | Mária      | Urbanová         |
| Urban        | Urban              | Mária      | Urbanová         |
| Dáša         | Nováková           | Patrícia   | Nováková         |
| Viera        | Silná              | Patrícia   | Nováková         |
| Ján          | Horváth            | Lucia      | Horváthová       |
| Zuzana       | Silná              | Viera      | Silná            |
| Zuzana       | Slabá              | Milena     | Slabá            |
| Zuzana       | Pravá              | Milena     | Slabá            |
| Zuzana       | Ľavá               | Milena     | Slabá            |
| Zuzana       | Stredná            | Milena     | Slabá            |

# NATURAL JOIN

- **NATURAL JOIN** je špeciálny typ implicitného spájania relácií. Inštruuje databázový RDBMS, aby autonómne analyzoval schémy oboch tabuliek, **vyhľadal všetky stĺpce s identickými názvami** a vykonal nad nimi **INNER JOIN**.
- Pri tomto type spojenia sa **striktne nepíše klauzula ON**. Spojovací predikát je zostavený automaticky na pozadí.
- **Architektonické varovanie pre prax (Anti-pattern):** V produkčnom prostredí sa používanie **NATURAL JOIN striktne zakazuje**. Spôsobuje dve kritické riziká:
  - **Sémantická slepota:** Spája tabuľky podľa názvu, nie podľa reálneho logického vzťahu.
  - **Extrémna krehkosť kódu (Fragility):** Ak administrátor v budúcnosti pridá do oboch tabuliek stĺpec s rovnakým názvom (napr. `upravene_dna` pre audit), dopyt sa začne automaticky spájať aj podľa tohto stĺpca, čím bez akéhokoľvek varovania prestane vracáť správne dáta.

# NATURAL JOIN

- SELECT

meno,  
o.priezvisko,  
v.od,  
v.do

FROM osoba AS o  
NATURAL JOIN vzťah AS v;  
-- 7 riadkov

| meno   | priezvisko | od                  | do                  |
|--------|------------|---------------------|---------------------|
| Adam   | Prvý       | 1937-06-01 00:00:00 | 1967-05-11 00:00:00 |
| Eva    | Prvá       | 1967-05-12 00:00:00 | 1988-07-22 00:00:00 |
| Zoly   | Múdry      | 1938-12-02 00:00:00 | 1965-03-11 00:00:00 |
| Nasťa  | Kováčová   | 1953-11-11 00:00:00 | NULL                |
| Jozef  | Urban      | 1970-07-22 00:00:00 | 1975-09-01 00:00:00 |
| Mária  | Urbanová   | 1980-03-04 00:00:00 | 2000-12-31 00:00:00 |
| Patrik | Novák      | 1997-07-31 00:00:00 | 2002-12-04 00:00:00 |

# Zhrnutie

| Typ spojenia | Relačný operátor / Logika v pamäti                                                          | Kardinalita výstupu nad OsobaDB                                                | Správanie k nezhodným n-ticiam                                                            | Praktické využitie                                                                                  |
|--------------|---------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| CROSS JOIN   | Kartézsky súčin relácií ( $\times$ ). Generuje kombinatorickú maticu.                       | 161 riadkov (Presný súčin množín: 23 osôb $\times$ 7 vzťahov)                  | Ignoruje akúkoľvek podmienku. Spojí bezhlavo každý riadok s každým.                       | Generovanie testovacích matíc, permutácií atribútov (napr. Kombinácie Produkt $\rightarrow$ Farba). |
| INNER JOIN   | Prienik množín ( $\bowtie$ ). Vyžaduje splnenie podmienky na oboch stranách.                | 7 riadkov (Presný počet platných väzieb vo Vzťah)                              | Nezhodné riadky sú kompletne eliminované (vypadnú všetky ženy a slobodní muži).           | Striktné párovanie entít s existenčnou závislosťou (napr. Faktúra $\rightarrow$ Odberateľ).         |
| LEFT JOIN    | Asymetrické zachovanie ľavej relácie ( $\bowtie\leftarrow$ ). Ľavá strana je dominantná.    | 24 riadkov (23 osôb + 1 duplicitný vzťah u Zolyho)                             | Nevyhodí nikoho zľava. Chýbajúci protipól zprava doplní hodnotou NULL.                    | Generovanie kompletných zoznamov a reportov (napr. Všetci študenti $\rightarrow$ ich známky).       |
| RIGHT JOIN   | Asymetrické zachovanie pravej relácie ( $\bowtie\rightarrow$ ). Pravá strana je dominantná. | 7 riadkov (Zhodný s INNER JOIN, nakoľko každý vzťah má u vás validačného muža) | Nevyhodí nič zprava. Ak by vľavo chýbala osoba, jej stĺpce sa vyplnia ako NULL.           | Zrkadlový ekvivalent LEFT JOINu (v čistom kódovom štýle sa mu odporúča vyhýbať).                    |
| FULL JOIN    | Zjednotenie množín ( $\cup$ ). Spojenie logiky LEFT a RIGHT JOIN.                           | 24 riadkov (V MySQL/MariaDB nutné emulovať cez operátor UNION)                 | Zachováva kompletne všetky riadky z oboch relácií, hluché miesta obojstranne dopĺňa NULL. | Komplexná auditná synchronizácia a hľadanie nekonzistentností v databázových skladoch.              |
| NATURAL JOIN | Implicitný prienik ( $\bowtie$ ) na základe zhody názvov stĺpce.                            | Fatálne zlyhanie logiky (Spojí Osoba.id = Vzťah.id, nie podľa cudzích kľúčov)  | Vyhodnotí sa ako INNER JOIN nad stĺpcami s rovnakým názvom. Ostatné riadky vymaže.        | Teoretické zjednodušenie zápisu, v praxi nebezpečné kvôli nečakanej zhode názvov.                   |