Vnořené dotazy - využití pro všechny DML příkazy, příklady použití operátorů IN, EXISTS, ALL a ANY, závislé vs nezávislé dotazy, vnořené dotazy vs spojení pomocí JOINs

vyrobek - id, nazev, cena, dodavatel_id
dodavatel - id, nazev
polozka - id, vyrobek_id, mnozstvi

Vnořené příkazy - subqueries, podselecty

- vždy pouze SELECT vnořený do jiného DML příkazu (SELECT, INSERT, UPDATE, DELETE)
- získání pomocných hodnot nebo množin, které hlavní dotaz dále využívá

- musí být vnořen v kulatých závorkách (Výjimka: INSERT … SELECT, závorka není potřeba)
- nesmí obsahovat JOIN
- může obsahovat ORDER BY jen pokud je použit s TOP/LIMIT

- poddotaz lze vnořit vícekrát -> jako první se provede ten nejvíce vnořený (max 1 až 2)
- u UPDATE a DELETE omezení množiny pro cizí tabulku -> není zde join
- u INSERT - vložení z množiny

V příkazu SELECT

- v klauzuli SELECT mezi atributy
  - musí vždy vracet jeho jednu hodnotu
  - př.: Počet dražších produktů než je tento produkt

```
SELECT p.nazev,
       p.cena,
       (SELECT Count(*)
        FROM   produkt p2
        WHERE  p2.cena > p.cena) AS pocet_drazsich
FROM   produkt p;
```

- v klauzuli FROM
  - poddotaz se chová jako virtuální tabulka
  - př.: průměrné ceny podle dodavatelů

```
SELECT t.dodavatel_id,
       t.prumer
FROM   (SELECT dodavatel_id,
               Avg(cena) AS prumer
        FROM   vyrobek
        GROUP  BY dodavatel_id) AS t;
```

- v klauzuli WHERE
  - nejčastější použití (IN, EXISTS, ALL, ANY, =, >)
  - př.: výrobky, které se objevují v položkách

```
SELECT v.nazev
FROM   vyrobek v
WHERE  v.id IN (SELECT p.vyrobek_id
                FROM   polozka p);
```

V příkazu UPDATE a DELETE

- v klauzuli WHERE
- Nahrazuje JOIN, protože UPDATE/DELETE běžně neumí joinovat tabulky
- př.: Zdražíme všechny výrobky dodavatelů, kteří mají průměrnou cenu > 100

```
UPDATE vyrobek
SET    cena = cena + 10
WHERE  dodavatel_id IN (SELECT dodavatel_id
                        FROM   vyrobek
```

```
                        GROUP  BY dodavatel_id
                        HAVING Avg(cena) > 100);
```

- př.: Smažeme všechny položky, které odkazují na výrobky s cenou pod 10

```
DELETE FROM polozka
WHERE  vyrobek_id IN (SELECT id
                      FROM   vyrobek
                      WHERE  cena < 10);
```

V příkazu INSERT

- za příkazem
- INSERT INTO Tabulka SELECT … FROM …
- př.:

```
INSERT INTO drahe_vyrobky(id, nazev)
SELECT id, nazev FROM   vyrobek
     WHERE  cena > 100;
```

Operátory ANY, ALL, EXISTS, IN

- porovnávání hodnot z hlavního dotazu s množinou výsledků z poddotazu
- tyto operátory nelze použít ve funkci CHECH() pro kontrolu integrity
- ANY, ALL se používají většinou pro porovnání hodnot (ne existenci -> IN, EXISTS)

ANY

- před operátorem musí být vždy matematický operátor (=, <, >, <=, >=, !=(<>))
- vrací true, pokud alespoň jedna hodnota z poddotazu splňuje podmínku
- WHERE <atribut> <operátor> ANY (poddotaz);

- př.: výrobky ze seznamu položek, které mají množství 10

```
SELECT v.nazev FROM   vyrobek v
WHERE  v.id = ANY (SELECT p.vyrobek_id
                   FROM   polozka p
                   WHERE  p.mnozstvi = 10
);
```

- př.: výrobky, které jsou dražší než průměr cen od libovolného dodavatele

```
SELECT vyrobek.nazev, vyrobek.cena FROM vyrobek
WHERE  vyrobek.cena > ANY(
      SELECT Avg(cena) FROM vyrobek
            GROUP BY dodavatel_id
);
```

ALL

- vrací true pokud všechny hodnoty podselectu splňují podmínku
- WHERE <atribut> <operátor> ALL (poddotaz);

- př.:

```
SELECT vyrobek.nazev FROM vyrobek
WHERE  vyrobek.id = ALL(
      SELECT Avg(cena)
      FROM   polozka položka.mnozstvi = 10
);
```

- př.: vyrobky s cenou > vsechny prumery

```
SELECT vyrobek.nazev, vyrobek.cena FROM vyrobek
      WHERE  vyrobek.cena > ALL(
            SELECT   Avg(cena) FROM vyrobek
            GROUP BY dodavatel_id
);
```

EXISTS

- Vrací TRUE, pokud poddotaz vrací alespoň jeden řádek
- Používá se často s korelovaným poddotazem.
- WHERE <atribut> EXISTS (poddotaz);
- př.: seznam dodavatelů, kteří mají alespoň jeden výrobek s cenou < 20

```
SELECT dodavatel.nazev
```

```
FROM   dodavatel
WHERE  EXISTS(SELECT vyrobek.nazev FROM vyrobek
              WHERE vyrobek.dodavatel_id = dodavatel.id
                    AND vyrobek.cena < 20
);
```

IN

- Porovnává hodnotu s množinou vrácenou podselectem - testuje členství v množině
- WHERE <atribut> IN (poddotaz);
- IN -> =ANY
- NOT IN -> <>ALL (!= ALL)

Závislé vs. nezávislé dotazy

Korelovaný dotaz

- nelze spustit samostatně
- obsahuje tabulku z předchozího dotazu

Nekorelovaný dotaz

- lze spustit samostatně
- nezávislý na hlavním dotazu

Vnořené dotazy vs spojení pomocí JOINs

Join

- pokud je jedna nebo více tabulek menších (tzn. číselníky) a jedna základní tabulka s velkým počtem záznamů (500 000 a více)

Podselect

- pokud máme tabulky přibližně stejně velké (500 000 a více)