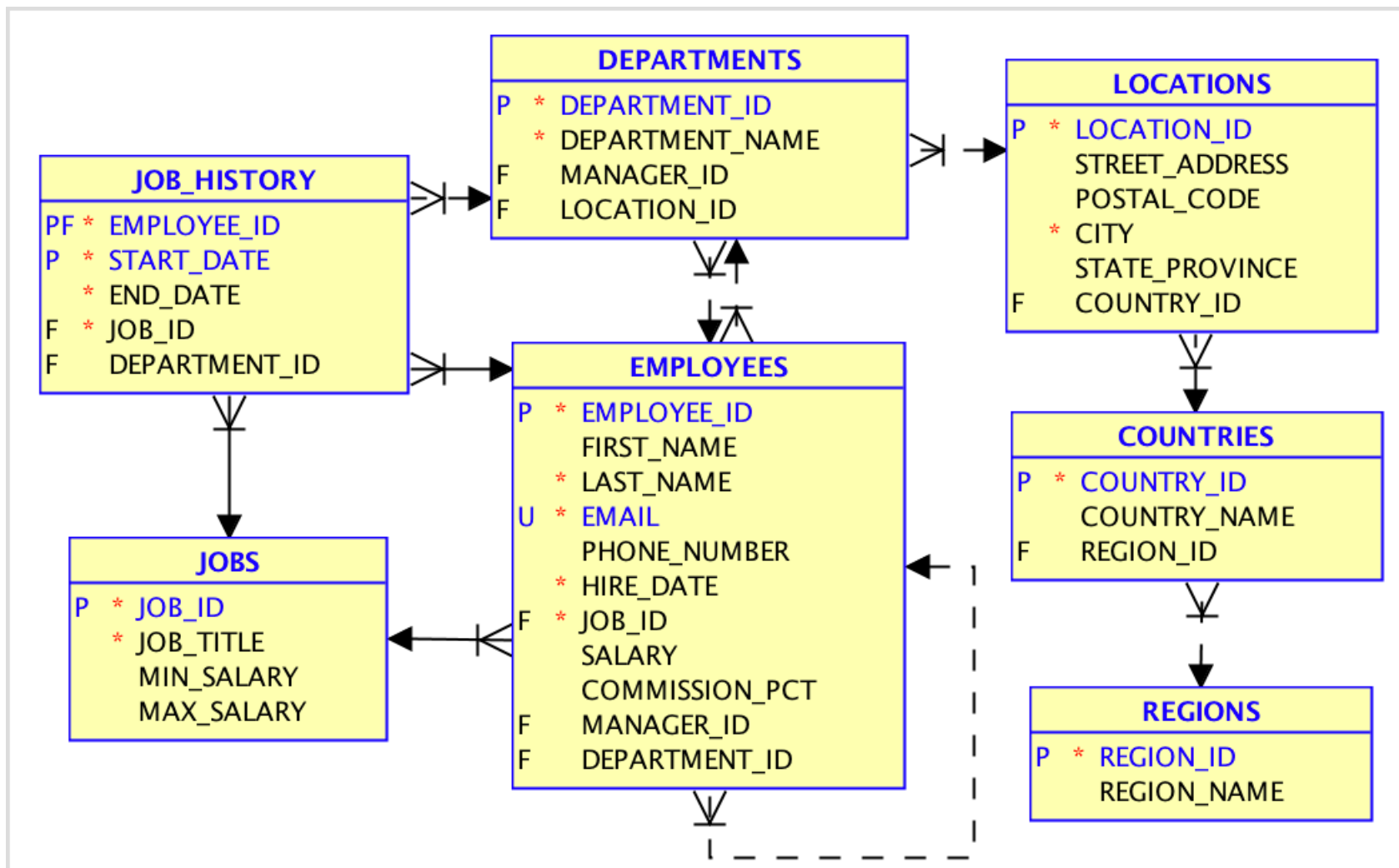


Databázové systémy

7. vnorené dopyty

HR schéma



Vnorené dotazy

- vnorený dotaz: podotázka, poddotaz, subquery, subselect
 - príkaz SELECT vo vnútri iného príkazu
- vonkajší dopyt - dopyt obsahujúci poddopyt

Príklad:

```
SELECT first_name, last_name FROM  
employees WHERE department_id=  
(SELECT department_id FROM employees  
WHERE employee_id=182);
```

Vnorené dotazy

- podľa toho, čo vráti vnorený príkaz **SELECT** na vyššiu úroveň rozlišujeme:
 - poddopyt vracajúci jednu hodnotu: jednoriadkové poddopyty
 - poddopyt vracajúci zoznam hodnôt: viacriadkové poddopyty

!výsledok musí zapadnúť do vonkajšieho dopytu

Vnorené dotazy

- Kde môžeme nájsť podotázky?
 - v klauzule WHERE a HAVING
 - v SELECT alebo FROM klauzule
 - v príkazoch INSERT, UPDATE, DELETE

Jednoriadkové podotázky

- vracia maximálne jednu hodnotu (inak NULL)
- podotázka v klauzule WHERE
 - Príklad:
 - zoznam zamestnancov, ktorí zarábajú najlepšie vo firme
 - najdlhšie pracujúci zamestnanec na aktuálnej pozícii

```
SELECT employee_id, first_name, last_name, salary
FROM employees
WHERE salary=(SELECT max(salary) FROM employees);
```

- podotázka v klauzule HAVING
 - Príklad:
 - zobrazte oddelenie a ich min plat, ktorý je väčší ako maximálny plat v oddelení 10

```
SELECT department_id, min(salary)
FROM employees
GROUP BY department_id
HAVING min(salary)>(SELECT max(salary) FROM employees
WHERE department_id=10);
```

Viacriadkové podotázky

- v klauzule WHERE je potrebné použiť jeden z nasledujúcich operátorov:
 - (NOT) IN: test existencie prvku v množine
 - IN / ANY
 - zobrazíť manažérov oddelení Human resources a Administration

```
SELECT employee_id, first_name, last_name, department_id
FROM employees
WHERE employee_id
IN (SELECT manager_id FROM departments WHERE
department_name IN ('Human Resources', 'Administration'));
```

- zobrazíť zamestnancov pracujúcich v meste Toronto a Seattle

Viacriadkové podotázky

- ANY: porovnáva, či hodnota atribútu spĺňa podmienku vzhľadom k aspoň jednému prvku množiny
 - >ANY znamená viac ako minimum,
 - použitím DISTINCT prejdeme viacnásobnému výberu riadku
 - zobrazte zamestnancov, ktorý zarábajú viac ako aspoň jeden zo zamestnancov oddelenia Administration

```
SELECT employee_id, first_name, last_name, salary
FROM employees
WHERE salary>ANY (SELECT salary FROM employees
WHERE department_id=(SELECT department_id FROM
departments
WHERE Department_name='Administration'));
```


Viacriadkové podotázky

- ALL: porovnáva, či hodnota atribútu spĺňa podmienku vzhľadom ku každému prvku množiny
 - >ALL znamená viac ako maximum
 - použitím DISTINCT prejdeme viacnásobnému výberu riadku
 - zoznam zamestnancov, ktorí zarábajú menej ako akýkoľvek zamestnanec oddelenia Administration

```
SELECT employee_id, first_name, last_name, salary
FROM employees
WHERE salary<ALL(SELECT salary FROM employees WHERE
department_id=(SELECT department_id FROM
departments WHERE department_name='Administration'))
ORDER BY 4;
```

Korelované podotázky

- podotázka, ktorá používa hodnotu z vonkajšieho dopytu
- nemôže byť vykonaná samostatne
- korelovaná podotázka sa vykonáva pre každý riadok z vonkajšieho dopytu – vykonovo náročná operácia

Kolerované podotázky

Príklad:

- zoznam zamestnancov, ktorých plat je vyšší ako priemerný plat v oddelení, v ktorom pracuje

```
SELECT first_name, last_name, salary, department_id
FROM employees e1
WHERE salary > (SELECT avg(salary) FROM employees
               WHERE department_id = e1.department_id);
```

- zoznam oddelení a počet zamestnancov

```
SELECT department_name, (SELECT count(*) FROM employees e
WHERE e.department_id = d.department_id) AS count_dep
FROM departments d;
```

Kolerované podotázky

- operátor EXIST (negácia NOT EXIST)
 - zisťuje, či výsledok podotázky obsahuje aspoň jeden záznam pre daný záznam vonkajšieho dopytu

zobrazte zoznam oddelení, ktoré majú/nemajú zamestnancov

```
SELECT department_name FROM departments d WHERE NOT  
EXISTS (SELECT 1 FROM employees e WHERE  
e.department_id=d.department_id);
```

Podotázky v klauzule SELECT a FROM

- inline view: podotázka v klauzule FROM
 - zobrazenie id zamestnanca, ktorý pracoval vo firme na najviac pracovných pozíciách

```
SELECT employee_id, count(*)+1  
FROM job_history GROUP BY employee_id HAVING count(*)=(SELECT  
max(Pocet) FROM (SELECT employee_id, count(*) POCET FROM  
job_history GROUP BY employee_id));
```

- podotazka v kaluzule SELECT:
 - zobrazit nazvy oddeleni a pocet zamestnancov pre jednotlivé oddelenia

```
SELECT department_name, (SELECT count(*) FROM employees e  
WHERE e.department_id=d.department_id) AS count_dep FROM  
departments d;
```

Vytvorenie tabuľky pomocou podotázky

Využitie:

- vytvorenie kópie tabuľky
- normalizácia/denormalizácia existujúcich tabuliek

Statement:

```
CREATE <meno tabuľky> AS subquery;
```

Príklad:

- vytvorte tabuľku emp_dep, ktorá bude obsahovať id a meno zamestnanca, plat, meno a id oddelenia(<50) pomocou tabuliek employees a departments

```
CREATE TABLE emp_dep AS  
(SELECT e.employee_id, e.first_name||' '||e.last_name name,  
e.salary, d.department_name, d.department_id FROM employees e  
LEFT OUTER JOIN departments d ON  
e.department_id=d.department_id WHERE d.department_id<50);
```

DML príkazy s podotázkou

- INSERT

INSERT INTO <meno tabuľky> (**subquery**);

Príklad:

- do tabuľky emp_dep vložte informácie o zamestnancoch oddelení s id >50

```
INSERT INTO emp_dep (SELECT e.employee_id,  
e.first_name||' '||e.last_name name, e.salary,  
d.department_name, d.department_id FROM employees  
e LEFT OUTER JOIN departments d ON  
e.department_id=d.department_id WHERE  
d.department_id>=50);
```

DML príkazy s podotázkou

- DELETE

```
DELETE FROM <meno tabuľky>  
WHERE subquery;
```

Príklad:

- zmazať všetkých zamestnancov, ktorý zarabajú menej ako priemerný plat

```
DELETE FROM employees WHERE salary<(SELECT avg(salary)  
FROM employees);
```

- zmazať oddelenia, ktoré nemajú zamestnanca

```
DELETE FROM departments WHERE department_id IN  
(SELECT department_id FROM departments d WHERE NOT EXISTS  
(SELECT 1 FROM employees e WHERE  
e.department_id=d.department_id));
```


DML príkazy s podotázkou

- UPDATE

```
UPDATE <meno tabuľky>  
SET <meno stĺpca1> = <výraz1>  
[,<meno stĺpca2> = <výraz2> ...]  
WHERE subquery;
```

```
UPDATE <meno tabuľky>  
SET (<meno stĺpca1>,..., <meno stĺpcaN>) =  
(SELECT <výraz1>,..., <výrazN> FROM ..)  
[WHERE <podmienka>];
```

Príklad:

- zvýšiť plat o 100 všetkým zamestnancom, ktorý majú plat nižší ako priemer

```
UPDATE employees SET salary=salary+100 WHERE salary<(SELECT avg(salary) FROM  
employees);
```

- nastaviť plat všetkým zamestnancom na hodnotu priemerného platu oddelenia, v ktorom pracujú

```
UPDATE employees e1 SET salary=(SELECT avg(salary) FROM employees e2 WHERE  
e2.department_id=e1.department_id);
```