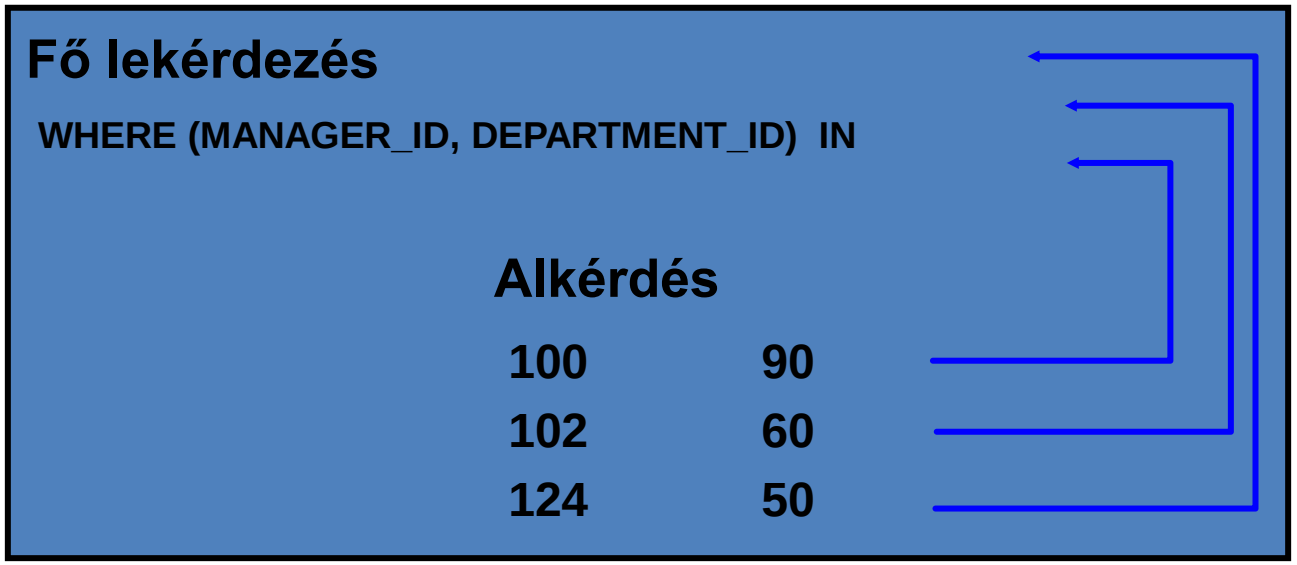
Alkérdés II.

Célkitűzések

- A lecke célja a következők bemutatása:
  - Több oszlopos alkérdések megadása
  - Skalár értéket visszaadó alkérdések használata
  - Korrelált alkérdések megadása
  - Az **`EXISTS`** és a **`NOT EXISTS`** operátorok
  - A **`WITH`** utasítás

Több oszlopos alkérdések



A fő lekérdezés sorai több soros és több oszlopos alkérdés eredményével kerülnek összehasonlításra.

Oszlopok összehasonlítása

- Az oszlopok összehasonlítása egy több oszlopos alkérdés esetén kétféleképpen történhet:
  - Páronkénti összehasonlítással
  - Nem páronkénti összehasonlítással

Páronkénti összehasonlítás

- Adjuk meg azoknak a dolgozóknak az adatait, akiknek ugyanaz a főnöke és ugyanazon az osztályon dolgoznak, mint a 199-es vagy a 174-es dolgozó azonosítóval rendelkező dolgozók valamelyike.

```
SELECT  employee_id, manager_id, department_id
FROM    employees
WHERE   (manager_id, department_id) IN
                        (SELECT manager_id, department_id
                         FROM   employees
                         WHERE  employee_id IN (199,174))
AND     employee_id NOT IN (199,174);
```

Nem páronkénti összehasonlítás

- Adjuk meg azoknak a dolgozóknak az adatait, akiknek ugyanaz a főnöke mint a 199-es vagy a 174-es dolgozó azonosítóval rendelkező dolgozók valamelyikének, és ugyanazon az osztályon dolgoznak, mint a 199-es vagy a 174-es dolgozó azonosítóval rendelkező dolgozók valamelyike.

```
SELECT   employee_id, manager_id, department_id
FROM     employees
WHERE    manager_id IN
                    (SELECT  manager_id
                     FROM    employees
                     WHERE   employee_id IN (174,199))
AND      department_id IN
                    (SELECT  department_id
                     FROM    employees
                     WHERE   employee_id IN (174,199))

AND     employee_id NOT IN(174,199);
```

Skalár értéket visszaadó alkérdések

- Az ilyen alkérdések pontosan egy soros és egy oszlopos eredményt adnak vissza, vagyis egy skalár értéket.
- Az ilyen alkérdések a következő helyeken is megadhatók:
  - A `DECODE` és `CASE` függvények feltétel és kifejezés részében
  - A `SELECT` utasítás minden részében, kivéve a `GROUP BY-t`

Példák skalár alkérdésre

– A CASE függvény kifejezés részében

```
SELECT employee_id, last_name,
       (CASE
        WHEN department_id =
                (SELECT department_id
                 FROM departments
                 WHERE location_id = 1800)
         THEN 'Canada' ELSE 'USA' END) location
FROM   employees;
```

20

- az ORDER BY részben

```
SELECT   employee_id, last_name
FROM     employees e
ORDER BY (SELECT department_name
          FROM departments d
          WHERE e.department_id = d.department_id);
```

Korrelált alkérdések

- A korrelált alkérdések a külső lekérdezés minden egyes sorára külön-külön végrehajtódnak.

Korrelált alkérdések

- Az alkérdés olyan oszlopra hivatkozik, amely a szülő (külső) lekérdezés valamelyik táblájában szerepel.

```
SELECT column1, column2, ...
FROM   table1   outer
WHERE  column1 operator
                      (SELECT   column1, column2
                       FROM     table2
                       WHERE    expr1 =
                                   outer.expr2);
```

Korrelált alkérdés használata

- Adjuk meg azokat a dolgozókat, akik többet keresnek a saját osztályuk átlagos fizetésénél.

```
SELECT last_name, salary, department_id
FROM   employees outer
WHERE  salary >
       (SELECT AVG(salary)
        FROM   employees
        WHERE  department_id =
        outer.department_id);
```

A külső lekérdezés minden sorára külön-külön végrehajtódik az alkérdés.

Korrelált alkérdés használata

Adjuk meg azon dolgozók adatait, akik legalább egyszer munkakört váltottak, vagyis legalább két külön sor szerepel róluk a JOB_HISTORY táblában.

```
SELECT e.employee_id, last_name,e.job_id
FROM   employees e
WHERE  2 <= (SELECT COUNT(*)
             FROM   job_history
             WHERE  employee_id = e.employee_id);
```

EMPLOYEE_ID	LAST_NAME	JOB_ID
101	Kochhar	AD_VP
176	Taylor	SA_REP
200	Whalen	AD_ASST

Az `EXISTS` operátor használata

- Az `EXISTS` operátor azt vizsgálja, hogy van-e egyáltalán sora az alkérdés eredményének.
- Ha az alkérdés eredményében előáll egy sor:
  - Az alkérdés végrehajtása nem folytatódik tovább
  - A feltétel a fő lekérdezésben **`TRUE`** (`IGAZ`) lesz
- Amíg az alkérdés nem eredményez egy sort sem:
  - A feltétel **`FALSE`** (`HAMIS`) lesz
  - Az alkérdés végrehajtása tovább folytatódik

Adjuk meg azokat a dolgozókat, akiknek van beosztottja

```
SELECT employee_id, last_name, job_id, department_id
FROM   employees outer
WHERE  EXISTS ( SELECT 'X'
                FROM   employees
                WHERE  manager_id =
                       outer.employee_id);
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90
103	Hunold	IT_PROG	60
108	Greenberg	FI_MGR	100
114	Raphaely	PU_MAN	30
120	Weiss	ST_MAN	50
121	Fripp	ST_MAN	50
122	Kaufling	ST_MAN	50
123	Vollman	ST_MAN	50
124	Mourgos	ST_MAN	50
145	Russell	SA_MAN	80
146	Partners	SA_MAN	80
147	Errazuriz	SA_MAN	80
148	Cambrault	SA_MAN	80
149	Zlotkey	SA_MAN	80
201	Hartstein	MK_MAN	20
205	Higgins	AC_MGR	110

18 rows selected.

Adjuk meg azokat az osztályokat, amelyeknek nincs dolgozója

```
SELECT department_id, department_name
FROM departments d
WHERE NOT EXISTS (SELECT 'X'
                  FROM   employees
                  WHERE  department_id = d.department_id);
```

DEPARTMENT_ID	DEPARTMENT_NAME
120	Treasury
130	Corporate Tax
140	Control And Credit
150	Shareholder Services
160	Benefits
170	Manufacturing
...	
260	Recruiting
270	Payroll

16 rows selected.

A `WITH` utasítás

- A `WITH` utasítás segítségével egy SELECT utasításban elnevezhetjük és többször is felhasználhatjuk ugyanazt a lekérdezés blokkot, ha az többször is előfordul egy összetett lekérdezésben.
- A `WITH` utasítás a lekérdezés blokkot végrehajtja, és annak eredményét átmenetileg eltárolja a felhasználó számára rendelkezésre álló tárterületen (temporary tablespace).
- A `WITH` utasítás javítja a lekérdezés hatékonyságát.

Példa `WITH` utasításra

- A `WITH` utasítás használatával adjunk meg egy lekérdezést, amely kiírja az osztály nevét, és az osztályon az fizetések összegét, de csak azokra az osztályokra, ahol ez az összegzett fizetés nagyobb, mint az ilyen osztályonként összegzett fizetések átlaga.

Példa `WITH` utasításra

```
WITH
dept_costs  AS (
    SELECT d.department_name, SUM(e.salary) AS dept_total
    FROM   employees e JOIN departments d
    ON     e.department_id = d.department_id
    GROUP BY d.department_name),
avg_cost    AS (
    SELECT SUM(dept_total)/COUNT(*) AS dept_avg
    FROM   dept_costs)
SELECT *
FROM   dept_costs
WHERE  dept_total >
         (SELECT dept_avg
          FROM avg_cost)
ORDER BY department_name;
```