Rendben, itt van 10 SQL lekérdezés a megadott adatokhoz, amelyek különböző **SQL kulcsszavakat** használnak a gyakorláshoz. Feltételezzük, hogy az adatok egy versenyzok nevű táblában vannak, és az oszlopnevek megegyeznek a megadottakkal (név, születési\_dátum, nemzetiség, rajtszám).

**SQL Lekérdezések Gyakorláshoz**

1. Összes versenyző nevének és nemzetiségének lekérdezése:

Ez egy egyszerű SELECT lekérdezés, ami az összes sorból kiválaszt bizonyos oszlopokat.

SQL

SELECT név, nemzetiség FROM versenyzok;

1. Az összes brit versenyző nevének lekérdezése:

Ez a lekérdezés a WHERE záradékot használja a szűrésre.

SQL

SELECT név FROM versenyzok WHERE nemzetiség = 'brit';

1. A német nemzetiségű versenyzők születési dátum szerint növekvő sorrendben:

Itt a WHERE és az ORDER BY kulcsszavakat használjuk.

SQL

SELECT név, születési\_dátum FROM versenyzok WHERE nemzetiség = 'német' ORDER BY születési\_dátum ASC;

1. Azoknak a versenyzőknek a neve és rajtszáma, akiknek van rajtszámuk (nem üres):

Ez a lekérdezés a WHERE záradékot használja a nem NULL értékek keresésére a rajtszám oszlopban.

SQL

SELECT név, rajtszám FROM versenyzok WHERE rajtszám IS NOT NULL;

1. Nemzetiségenként hány versenyző van?:

Ez a lekérdezés a GROUP BY és a COUNT aggregáló függvényt használja.

SQL

SELECT nemzetiség, COUNT(név) AS versenyzok\_szama FROM versenyzok GROUP BY nemzetiség;

1. A legfiatalabb versenyző neve és születési dátuma:

Itt a MAX aggregáló függvényt használjuk a születési\_dátum oszlopra (a legkésőbbi dátum a legfiatalabb).

SQL

SELECT név, születési\_dátum FROM versenyzok ORDER BY születési\_dátum DESC LIMIT 1;

-- Alternatív, ha a MAX-ot szeretnénk használni:

-- SELECT név, születési\_dátum FROM versenyzok WHERE születési\_dátum = (SELECT MAX(születési\_dátum) FROM versenyzok);

*Megjegyzés:* A LIMIT 1 gyakran hatékonyabb a "legfiatalabb/legidősebb" típusú lekérdezésekre. Az alternatív megoldás a szub-lekérdezéssel is működik, és jobban szemlélteti a MAX használatát.

1. Azoknak a nemzetiségeknek a listája, amelyekből egynél több versenyző van:

Ez a lekérdezés a GROUP BY és a HAVING záradékot használja, ami a GROUP BY után szűr.

SQL

SELECT nemzetiség FROM versenyzok GROUP BY nemzetiség HAVING COUNT(név) > 1;

1. A rajtszámmal rendelkező versenyzők átlagos életkora (feltételezve, hogy a születési dátumból számolható az életkor, vagy egy külön életkor oszlop létezik):

Mivel nincs életkor oszlop, és a pontos életkor számítása bonyolultabb SQL-függvényeket igényelne, ami kívül esik a feladat hatókörén, ezért egy egyszerűsített megközelítést használunk: megszámoljuk a rajtszámmal rendelkező versenyzők számát nemzetiségenként.

Ha feltételezzük, hogy van egy 'életkor' oszlop:

SQL

-- SELECT AVG(életkor) FROM versenyzok WHERE rajtszám IS NOT NULL;

*Az eredeti adatokhoz illeszkedve, és a COUNT gyakorlására:*

SQL

SELECT nemzetiség, COUNT(rajtszám) AS rajtszam\_szam FROM versenyzok WHERE rajtszám IS NOT NULL GROUP BY nemzetiség;

1. A legmagasabb rajtszámú versenyző neve és nemzetisége:

Ez a lekérdezés a MAX aggregáló függvényt használja.

SQL

SELECT név, nemzetiség, rajtszám FROM versenyzok WHERE rajtszám = (SELECT MAX(CAST(rajtszám AS UNSIGNED)) FROM versenyzok WHERE rajtszám IS NOT NULL);

*Megjegyzés:* A CAST(rajtszám AS UNSIGNED) szükséges lehet, ha a rajtszám oszlop nem numerikus típusú, és számként akarjuk kezelni a maximum keresésénél. Ha a rajtszám oszlop integer típusú, akkor a CAST nem szükséges.

1. Az összes versenyző neve, nemzetisége és születési dátuma, a nemzetiség és azon belül a születési dátum szerint csökkenő sorrendben:

Ez a lekérdezés az ORDER BY-t használja több oszlopra.

SQL

SELECT név, nemzetiség, születési\_dátum FROM versenyzok ORDER BY nemzetiség ASC, születési\_dátum DESC;

# SQL Lekérdezések Gyakorlása

Az alábbiakban 10 SQL lekérdezést talál, amelyekkel gyakorolhatod a SELECT, WHERE, GROUP BY, ORDER BY, MAX, AVG és COUNT függvények használatát az adott versenyző adatokon.

## 1. Összes versenyző listázása

```sql

SELECT \* FROM versenyzok;

```

## 2. Brit versenyzők név és rajtszáma

```sql

SELECT nev, rajtszam

FROM versenyzok

WHERE nemzetiseg = 'brit'

ORDER BY nev;

```

## 3. Nemzetiségek szerinti versenyzők száma

```sql

SELECT nemzetiseg, COUNT(\*) as darab

FROM versenyzok

GROUP BY nemzetiseg

ORDER BY darab DESC;

```

## 4. Legidősebb versenyző(ek) adatai

```sql

SELECT nev, szuletesi\_datum, nemzetiseg

FROM versenyzok

WHERE szuletesi\_datum = (SELECT MIN(szuletesi\_datum) FROM versenyzok);

```

## 5. Átlagos születési év nemzetiségenként

```sql

SELECT nemzetiseg, AVG(YEAR(szuletesi\_datum)) as atlag\_szuletesi\_ev

FROM versenyzok

GROUP BY nemzetiseg

ORDER BY atlag\_szuletesi\_ev;

```

## 6. Versenyzők száma, akiknek nincs rajtszáma

```sql

SELECT COUNT(\*) as nincs\_rajtszam

FROM versenyzok

WHERE rajtszam IS NULL OR rajtszam = '';

```

## 7. Legnépszerűbb rajtszám (legtöbb versenyzővel)

```sql

SELECT rajtszam, COUNT(\*) as darab

FROM versenyzok

WHERE rajtszam IS NOT NULL AND rajtszam != ''

GROUP BY rajtszam

ORDER BY darab DESC

LIMIT 1;

```

## 8. 1980 után született versenyzők nemzetiségenként

```sql

SELECT nemzetiseg, COUNT(\*) as darab

FROM versenyzok

WHERE YEAR(szuletesi\_datum) > 1980

GROUP BY nemzetiseg

ORDER BY darab DESC;

```

## 9. 10 legfiatalabb versenyző

```sql

SELECT nev, szuletesi\_datum, nemzetiseg

FROM versenyzok

ORDER BY szuletesi\_datum DESC

LIMIT 10;

```

## 10. Nemzetiségek, ahol az átlagéletkor 40 év felett van

```sql

SELECT nemzetiseg, AVG(YEAR(CURRENT\_DATE) - YEAR(szuletesi\_datum)) as atlageletkor

FROM versenyzok

GROUP BY nemzetiseg

HAVING atlageletkor > 40

ORDER BY atlageletkor DESC;

```

Ezek a lekérdezések lehetővé teszik az adatok különböző szempontú elemzését és a SQL alapvető függvényeinek gyakorlását. A lekérdezések között szerepel szűrés (WHERE), csoportosítás (GROUP BY), rendezés (ORDER BY), valamint aggregáló függvények (COUNT, AVG, MAX) használata is.