МӘЛІМЕТТЕР ҚОРЫ · ДӘРІС 3

Негізгі SQL командалары

Шектеулер · сұрыптау · агрегаттау

```
SELECT name
FROM   students
WHERE  gpa > 3.5
ORDER BY name;
```

Оқытушы: Айдана Карибаева

01

SQL тілінің бөліктері

SQL (Structured Query Language) — реляциялық МҚ-мен жұмыс істеуге арналған стандартты тіл.

DDL

деректерді **анықтау**: құрылымды құру, өзгерту

`CREATE` `ALTER` `DROP` `TRUNCATE`

DML

деректерді **өңдеу**: таңдау, қосу, өзгерту, жою

`SELECT` `INSERT` `UPDATE` `DELETE`

DCL

деректерді **басқару**: рұқсаттар мен құқықтар

`GRANT` `REVOKE`

TCL

транзакцияларды басқару

`COMMIT` `ROLLBACK` `SAVEPOINT`

02

SELECT сұранысының құрылымы

SELECT бағандар — ⑤ не шығару

FROM кесте — ① қайдан

WHERE шарт — ② жолдарды сүзу

GROUP BY бағаң — ③ топтау

HAVING шарт — ④ топтарды сүзу

ORDER BY бағаң — ⑥ сұрыптау

Жазу реті мен **орындалу реті** әртүрлі: МҚБЖ алдымен FROM, соңында ORDER BY орындайды.

> Сұраныс нәтижесі — жаңа **уақытша реляция** (кесте).

03 SELECT және DISTINCT

Customers

CUSTOMERNAME	CITY	COUNTRY
Alfreds	Berlin	Germany
Ana Trujillo	México	Mexico
Antonio	México	Mexico
Around the Horn	London	UK

```
SELECT CustomerName, City
FROM Customers;
```

```
SELECT * FROM Customers;
```

* — барлық бағандар

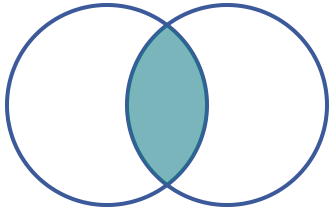
```
SELECT DISTINCT Country
FROM Customers;
```

COUNTRY
Germany
Mexico
UK

тек **әртүрлі** мәндер — Mexico бір рет

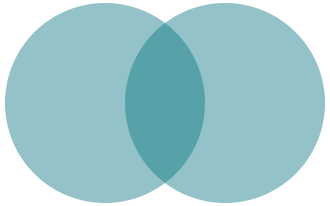
04

WHERE және логикалық операторлар



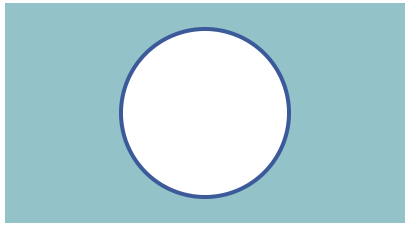
AND

барлық шарт орындалса



OR

кемінде бір шарт орындалса



NOT

шарт орындалмаса

```
SELECT EmpName FROM EMPLOYEE
WHERE DeptID = 9 AND SalaryCode <= 23;
```

```
SELECT EmpName FROM EMPLOYEE
WHERE DeptID < 7 OR DeptID > 12;
```

05

BETWEEN және LIKE

BETWEEN — мәндер ауқымы

```
SELECT EmpName FROM EMPLOYEE
WHERE SalaryCode BETWEEN 10 AND 45;
```



шекаралар **қоса** есептеледі

LIKE — үлгі бойынша іздеу

ТАҢБА	МАҒЫНАСЫ	МЫСАЛ
`%`	кез келген сан таңба (0 да)	`'La%'`
`_`	дәл бір таңба	`'_r%'`
`[ ]`	жақшадағы бір таңба	`'[bsp]%'`
`^`	жақшадан тыс таңба	`'[^a]%'`
`-`	таңбалар ауқымы	`'[a-c]%'`

06

NULL мәні

NULL — белгісіз мән. Ол 0 емес және бос жол емес. Сондықтан `= NULL` емес, `IS NULL` қолданылады.

CUSTOMERNAME	ADDRESS
Alfreds	Obere Str. 57
Ana Trujillo	*NULL*
Antonio	Mataderos 2312
Berglunds	*NULL*

```sql
SELECT CustomerName
FROM Customers
WHERE Address IS NULL;
```

→ Ana Trujillo, Berglunds

```sql
SELECT CustomerName
FROM Customers
WHERE Address IS NOT NULL;
```

→ Alfreds, Antonio

07 Шектеулер (constraints)

Кестедегі деректерге қойылатын **ережелер**.

ШЕКТЕУ	НЕ ҚАМТАМАСЫЗ ЕТЕДІ	МЫСАЛ
NOT NULL	баған бос болмайды	`name VARCHAR(50) NOT NULL`
UNIQUE	барлық мәндер бірегей	`email VARCHAR(255) UNIQUE`
CHECK	мән шартты қанағаттандырады	`CHECK (age >= 18)`
DEFAULT	мән берілмесе — әдепкі мән	`city VARCHAR(30) DEFAULT 'Almaty'`
PRIMARY KEY	NOT NULL + UNIQUE, жолды анықтайды	`id INT PRIMARY KEY`
FOREIGN KEY	басқа кестемен байланыс	`REFERENCES DEPARTMENT(DeptID)`

08 ALTER TABLE: бар кестеге шектеу қосу

Алғашқы кілт

```
ALTER TABLE EMPLOYEE
  ADD CONSTRAINT Emp_PK PRIMARY KEY(EmpID);
```

Құрама кілт

```
ALTER TABLE EMP_SKILL
  ADD CONSTRAINT EmpSkill_PK PRIMARY KEY(EmpID, SkillID);
```

Сыртқы кілт

```
ALTER TABLE EMPLOYEE
  ADD CONSTRAINT Emp_FK FOREIGN KEY(DeptID)
  REFERENCES DEPARTMENT(DeptID);
```

09

INSERT, UPDATE, DELETE

INSERT — қосу

```
INSERT INTO EMPLOYEE
VALUES(91, 'Smither', 12);
```

сандық емес мәндер '' ішінде

UPDATE — өзгерту

```
UPDATE Customers
SET ContactName =
  'Alfred Schmidt'
WHERE CustomerID = 1;
```

DELETE — жою

```
DELETE FROM Customers
WHERE CustomerName =
  'Alfreds';
```

⚠ UPDATE мен DELETE-те **WHERE** болмаса, кестедегі **барлық** жазбалар өзгереді немесе жойылады!

10 ORDER BY және TOP

```
SELECT * FROM Products
ORDER BY Price DESC;
```

PRODUCT	PRICE ↓
Côte de Blaye	263.5
Thüringer	123.8
Mishi Kobe	97.0
Chai	18.0

Сұрыптау

ASC	өсу ретімен (әдепкі)
DESC	кему ретімен

```
ORDER BY Country, CustomerName;
```

алдымен елі, сосын аты бойынша

TOP — жазбалар саны

```
SELECT TOP 3 *
FROM Customers;
```

мыңдаған жазбасы бар үлкен кестелерде пайдалы

11 Агрегаттық функциялар

PROJECT

PROJID	HOURS
8	120
9	80
10	200
11	100

COUNT
4
жолдар саны

SUM
500
қосынды

AVG
125
орташа

MIN
80
ең кіші

MAX
200
ең үлкен

```
SELECT MIN(Hours) AS MinimumHours,
       MAX(Hours) AS MaximumHours,
       AVG(Hours) AS AverageHours
FROM PROJECT WHERE ProjID > 7;
```

12 GROUP BY және HAVING

EMPLOYEE

Dept	Emp
IT	Әлі
IT	Дана
HR	Ерлан
Sales	Мира
Sales	Нұр
Sales	Олжас

GROUP BY →

Dept	COUNT(*)
IT	2
~~HR~~	~~1~~
Sales	3

HAVING COUNT(*) > 1
→ HR тобы алынып тасталады

```sql
SELECT Dept, COUNT(*)
FROM EMPLOYEE
GROUP BY Dept
HAVING COUNT(*) > 1;
```

WHERE	топтауға **дейін** жолдарды сүзеді
HAVING	топтаудан **кейін** топтарды сүзеді