Әл-Фараби атындағы Қазақ ұлттық университеті
Медицина және денсаулық сақтау факультеті
Эпидемиология, биостатистика және дәлелді медицина кафедрасы

ДӘРІС 7

Деректер қорлары және сұраныс логикасы: кестелер, кілттер, қарапайым SQL сұраныстары

Пән: Денсаулық сақтаудағы ақпараттық технологиялар

Лектор:
Абдрахманова Зинат Батырхановна
Эпидемиология, биостатистика және дәлелді медицина кафедрасының доценті
Медицина және денсаулық сақтау факультеті, Әл-Фараби атындағы Қазақ ұлттық университеті

Алматы, 2026

Сабақтың мақсаты және күтілетін нәтижелер

МАҚСАТЫ

Реляциялық деректер қорларының негізгі ұғымдарын, кестелер арасындағы байланыстарды түсіндіру және денсаулық сақтау деректерінен ақпарат алуға арналған қарапайым SQL сұраныстарын құрастыру дағдысын қалыптастыру.

Күтілетін нәтижелер: сабақ соңында білім алушы...

1. деректер қоры мен деректер қорын басқару жүйесі (ДҚБЖ) ұғымдарын түсіндіреді;
2. бастапқы және сыртқы кілттерді анықтап, кестелер арасындағы байланыс түрлерін ажыратады;
3. нормализацияның негізгі мақсатын түсіндіреді;
4. SELECT, WHERE, ORDER BY, GROUP BY операторларымен сұраныстар жазады;
5. JOIN арқылы бірнеше кестедегі деректерді біріктіреді.

Деректер қоры және ДҚБЖ

Негізгі ұғымдар

- **Деректер қоры (ДҚ)** — өзара байланысты, құрылымдалған деректердің ұйымдастырылған жиынтығы
- **ДҚБЖ** — деректерді құру, сақтау, өзгерту және іздеуге арналған бағдарлама
- **Мысалдар:** PostgreSQL, MySQL, Microsoft SQL Server, Oracle, SQLite, MS Access
- **SQL** (Structured Query Language) — деректер қорымен жұмыс істеудің стандартты тілі

Электрондық кестеден айырмашылығы

- Миллиондаған жазбалармен тиімді жұмыс
- Бірнеше пайдаланушының бір мезгілде жұмыс істеуі
- Деректер тұтастығы: типтер, шектеулер, кілттер
- Қол жеткізу құқықтарын басқару және аудит
- МАЖ, тіркелімдер мен ЭДП деректер қорларына негізделген

Реляциялық модель: мысал схемасы (шартты деректер)

Кесте	Өрістер (бағандар)	Мазмұны
patients	**patient_id (PK)**, sex, birth_date, district	Пациенттер туралы бір реттік деректер
visits	**visit_id (PK)**, *patient_id (FK)*, visit_date, clinic	Әр қаралу — жеке жазба
diagnoses	**diag_id (PK)**, *visit_id (FK)*, icd10	Қаралу кезінде қойылған диагноздар
lab_results	**lab_id (PK)**, *visit_id (FK)*, test, value, unit	Зертханалық нәтижелер

Кесте (қатынас) — жолдардан (жазбалар) және бағандардан (өрістер/атрибуттар) тұрады. Кестелер кілттер арқылы байланысады: patients → visits → diagnoses.

Кілттер және кестелер арасындағы байланыстар

Бастапқы кілт (Primary Key, PK)

Кестедегі әр жазбаны бірегей анықтайтын өріс; қайталанбайды және бос болмайды (мысалы, patient_id).

Сыртқы кілт (Foreign Key, FK)

Басқа кестенің бастапқы кілтіне сілтеме жасайтын өріс (visits.patient_id → patients.patient_id).

Байланыс түрлері

1:1 — пациент ↔ ЭДП; **1:N** — бір пациенттің көп қаралуы; **M:N** — дәрігерлер ↔ пациенттер (аралық кесте арқылы).

Сілтемелік тұтастық

Сыртқы кілт тек бар жазбаға сілтеме жасай алады: жоқ пациентке қаралу тіркелмейді.

Нормализация негіздері

- **Мақсаты:** деректердің қайталануын болдырмау және өзгерту кезіндегі қайшылықтардың (аномалиялардың) алдын алу.
- **Нормаланбаған кесте мысалы:** әр қаралу жолында пациенттің аты, мекенжайы, туған күні қайта жазылады — мекенжай өзгерсе, барлық жолды түзету қажет.
- **1NF:** әр ұяшықта бір ғана (атомарлы) мән; қайталанатын топтар жоқ.
- **2NF:** барлық өрістер бастапқы кілтке толық тәуелді.
- **3NF:** өрістер бір-біріне емес, тек кілтке тәуелді (мысалы, аудан атауы жеке анықтамалық кестеде).
- **Нәтиже:** пациент туралы деректер — бір кестеде, бір рет; қаралулар мен диагноздар — байланысқан жеке кестелерде.

SQL: деректерді таңдау және сүзу

Сұраныс құрылымы: **SELECT** (қандай бағандар) → **FROM** (қай кестеден) → **WHERE** (қандай шарт) → **ORDER BY** (сұрыптау).

```sql
-- 1. Барлық әйел пациенттер
SELECT patient_id, birth_date, district
FROM patients
WHERE sex = 'F';

-- 2. 2026 жылғы қаралулар, күні бойынша
SELECT visit_id, patient_id, visit_date
FROM visits
WHERE visit_date >= '2026-01-01'
ORDER BY visit_date DESC;
```

- **Салыстыру:** =, <>, >, <, >=, <=
- **Логикалық:** AND, OR, NOT
- **Тізім / ауқым:** IN (...), BETWEEN ... AND ...
- **Үлгі:** LIKE 'I1%' — I1-ден басталатын барлық кодтар
- **Бос мәндер:** IS NULL / IS NOT NULL

SQL: агрегаттау және топтастыру

Агрегаттық функциялар: COUNT (саны), SUM (қосынды), AVG (орташа), MIN, MAX. **GROUP BY** — топтар бойынша есептеу, **HAVING** — топтарды сүзу.

```sql
-- Гипертензиялық аурулар (I10–I15) саны
-- аудандар бойынша
SELECT p.district,
       COUNT(*) AS n_cases
FROM diagnoses d
JOIN visits v   ON d.visit_id = v.visit_id
JOIN patients p ON v.patient_id = p.patient_id
WHERE d.icd10 LIKE 'I1%'
GROUP BY p.district
HAVING COUNT(*) >= 10
ORDER BY n_cases DESC;
```

- Нәтиже: әр аудан үшін бір жол және жағдайлар саны.
- WHERE — жолдарды топтастыруға дейін сүзеді.
- HAVING — топтастырудан кейін сүзеді.
- AS — нәтиже бағанына атау береді.

SQL: кестелерді біріктіру (JOIN)

```sql
-- 2 типті қант диабеті (E11) бар пациенттер және олардың HbA1c нәтижелері
SELECT p.patient_id, p.sex, v.visit_date, l.value AS hba1c
FROM patients p
JOIN visits v           ON p.patient_id = v.patient_id
JOIN diagnoses d        ON v.visit_id = d.visit_id
LEFT JOIN lab_results l ON v.visit_id = l.visit_id AND l.test = 'HbA1c'
WHERE d.icd10 = 'E11';
```

- **INNER JOIN** — екі кестеде де сәйкес жазбасы бар жолдарды ғана қайтарады.
- **LEFT JOIN** — сол жақ кестенің барлық жолдарын қайтарады; оң жақта сәйкестік болмаса — NULL (мысалы, HbA1c тапсырмаған пациенттер де көрінеді).
- Кесте бүркеншік аттары (p, v, d, l) сұранысты қысқа әрі түсінікті етеді.

Қорытынды, бақылау сұрақтары және әдебиеттер

Негізгі тұжырымдар

- Реляциялық ДҚ деректерді байланысқан кестелерде, қайталаусыз сақтайды.
- Бастапқы кілт жазбаны бірегей анықтайды, сыртқы кілт кестелерді байланыстырады.
- SELECT–FROM–WHERE–GROUP BY–ORDER BY — сұраныстың негізгі логикасы.
- JOIN бірнеше кестедегі клиникалық деректерді біріктіріп талдауға мүмкіндік береді.

Бақылау сұрақтары

1. Деректер қорының электрондық кестеден артықшылықтары қандай?
2. Бастапқы және сыртқы кілттің айырмашылығын мысалмен түсіндіріңіз.
3. WHERE мен HAVING операторларының айырмашылығы неде?
4. 60 жастан асқан пациенттер санын аудандар бойынша есептейтін сұраныс жазыңыз.

Ұсынылатын әдебиеттер: 1. Magnuson J.A., Dixon B.E. Public Health Informatics and Information Systems. 4th ed. Springer, 2020. 2. Hardy L. et al. Health Informatics: An Interprofessional Approach. 3rd ed. Elsevier, 2023. 3. PostgreSQL Documentation: SQL Tutorial. https://www.postgresql.org/docs/