

Veri Tabanı Sistemleri

Uygulama Geliştirme

H. Turgut Uyar Şule Öğüdücü

2002-2012

1 / 65

License



©2002-2012 T. Uyar, Ş. Öğüdücü

You are free:

- ▶ to Share – to copy, distribute and transmit the work
- ▶ to Remix – to adapt the work

Under the following conditions:

- ▶ Attribution – You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- ▶ Noncommercial – You may not use this work for commercial purposes.
- ▶ Share Alike – If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

Legal code (the full license):

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

2 / 65

Konular

Uygulama Geliştirme

Giriş
Gömülü SQL
ODBC
JDBC

SQL

Kayıtlı Yordamlar
Görüntüler
İzinler
İpuçları

3 / 65

Giriş

- ▶ veri tabanı dili ile genel amaçlı bir programlama dilinin birlikte kullanımı
- ▶ uygulamanın yazıldığı genel amaçlı dil: **taban dil**
- ▶ SQL ile taban dil arasında uyumsuzluk:
 - ▶ SQL işlemleri kümeler üzerinde
 - ▶ genel amaçlı dillerde yineleme yapıları

4 / 65

Program Yapısı

- ▶ bağlantı kur
 - ▶ sunucu, veri tabanı, kullanıcı adı, parola
- ▶ gerekikçe komut çalıştır:
 - ▶ güncelleme işlemleri işlemde etkilenen satır sayısını döndürür
 - ▶ sorgulama işlemleri sonuç kümeleri döndürür
→ döngüyle satır satır gez
- ▶ bağlantıyı kopar

5 / 65

Uygulama Geliştirme Yöntemleri

- ▶ uygulama programı arayüzü (API)
- ▶ gömülü SQL
- ▶ ODBC
- ▶ dil standart arayüzleri

6 / 65

Uygulama Programı Arayüzü

- ▶ SQL sunucusunun kitaplık fonksiyonlarını çağırarak
- ▶ avantajı: hızlı
- ▶ dezavantajı: sunucudan sunucuya değişiyor

7 / 65

Uygulama Programı Arayüzü Örneği

Örnek (PostgreSQL - C)

```
#include <libpq-fe.h>

int main(int argc, char *argv[])
{
    /* connect */
    /* execute query */
    /* disconnect */
}
```

8 / 65

Uygulama Programı Arayüzü Örneği

Örnek (veri tabanı bağlantısı)

```
/* PGconn *conn; */

conn = PQconnectdb("host=localhost dbname=imdb"
                  " user=itucs password=itucs");
if (PQstatus(conn) == CONNECTION_BAD) {
    fprintf(stderr, "Connection failed.\n");
    exit(1);
}
/* execute query */
PQfinish(conn);
```

9 / 65

Uygulama Programı Arayüzü Örneği

Örnek (sorgunun işlenişi)

```
/* PGresult *result; */

sprintf(query, "SELECT TITLE, SCORE"
            " FROM MOVIE WHERE (YR = %d)", year);
result = PQexec(conn, query);
if (PQresultStatus(result) != PGRES_TUPLES_OK) {
    fprintf(stderr, "Query failed.\n");
    PQclear(result);
    PQfinish(conn);
    exit(1);
}
```

10 / 65

Uygulama Programı Arayüzü Örneği

Örnek (sonuç kümesinin işlenişi)

```
for (i = 0; i < PQntuples(result); i++) {
    title = PQgetvalue(result, i, 0);
    score = PQgetvalue(result, i, 1);
    ...
}

PQclear(result);
```

11 / 65

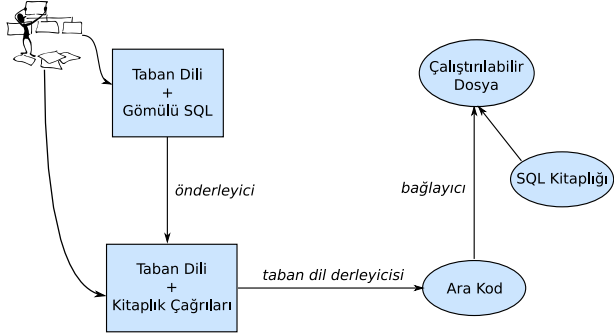
Gömülü SQL

- ▶ aşamalar:
 1. taban dil içinde SQL komutları işaretlenir: **EXEC SQL**
 2. gömülü SQL önışleyicisi:
gömülü SQL komutları → uygulama programı arayüzü çağırılır
 3. taban dil derleyicisi
- ▶ avantajları: hızlı, standart
- ▶ dezavantajları: kullanışsız, farklı diller için desteği yok

▶ Gömülü SQL bölümünü atla

12 / 65

Gömülü SQL



13 / 65

Gömülü SQL Standardı

- ▶ taban dil ile değişken paylaşımı
- ▶ hata denetimi
- ▶ sorgu sonuçlarının uyarlanması

14 / 65

Değişken Paylaşımı

Yazım

```
EXEC SQL BEGIN DECLARE SECTION;
shared variables
EXEC SQL END DECLARE SECTION;
```

- ▶ SQL komutlarında taban dil değişkenlerinin önüne ':'

15 / 65

Hata Denetimi

Hata İşleme

```
EXEC SQL WHENEVER
{ SQLERROR | SQLWARNING | NOT FOUND }
{ STOP | CONTINUE | DO command | GOTO label }
```

16 / 65

Sorgu Sonuçlarının Uyarlanması

İmleçler

```
EXEC SQL DECLARE cursor_name CURSOR FOR
SELECT ...;
EXEC SQL OPEN cursor_name;
EXEC SQL FETCH IN cursor_name INTO variables;
EXEC SQL CLOSE cursor_name;
```

- ▶ tanımlama işleminde sorgu çalıştırılmaz
- ▶ açma işlemiyle sorgu çalıştırılır
 - ▶ imleç sonuç kümesinin ilk satırına konumlandırılır

17 / 65

Gömülü SQL Örneği

Örnek (veri tabanına bağlanma)

```
EXEC SQL BEGIN DECLARE SECTION;
int year;
char *title = NULL, *score = NULL;
EXEC SQL END DECLARE SECTION;

EXEC SQL CONNECT TO imdb
USER itucs IDENTIFIED BY itucs;

/* process query */

EXEC SQL DISCONNECT;
```

18 / 65

Gömülü SQL Örneği

Örnek (sorgunun işlenmesi)

```
scanf("%d", &year);
EXEC SQL DECLARE c_query CURSOR FOR
    SELECT TITLE, SCORE FROM MOVIE
    WHERE (YR = :year);
EXEC SQL OPEN c_query;

/* execute query */

EXEC SQL CLOSE c_query;
EXEC SQL COMMIT;
```

19 / 65

Gömülü SQL Örneği

Örnek (sorgunun çalıştırılması)

```
EXEC SQL WHENEVER NOT FOUND DO break;
while (1) {
    EXEC SQL FETCH c_query INTO :title, :score;
    ...
}
```

20 / 65

ODBC

- ▶ **ODBC**: Open DataBase Connectivity
uygulama ile sunucu arasında bir servis katmanı
- ▶ avantajları: standart
- ▶ dezavantajları: yavaş

21 / 65

ODBC Mimarisi

- ▶ uygulama
- ▶ sürücü yöneticisi
 - ▶ ODBC sürücülerini kaydeder
 - ▶ uygulamadan gelen ODBC isteklerini ilgili sürücüye aktarır
- ▶ sürücü
 - ▶ istekleri veri kaynağına uygun şekilde iletir
- ▶ veri kaynağı
 - ▶ sürücüden gelen komutları işler

22 / 65

ODBC Örneği

Örnek (PHP)

```
$conn = odbc_connect("imdb", "itucs", "itucs");
$query = "SELECT TITLE, SCORE FROM MOVIE"
        . " WHERE (YR = " . $year . ")";
$result = odbc_exec($conn, $query);

/* process the result set */

odbc_close($conn);
```

23 / 65

ODBC Örneği

Örnek (sonuç kümesinin işlenmesi)

```
echo "<table>\n";
while (odbc_fetch_row($result)) {
    $title = odbc_result($result, "title");
    $score = odbc_result($result, "score");
    echo "<tr>\n";
    echo "    <td>$title</td>\n";
    echo "    <td>$score</td>\n";
    echo "</tr>\n";
}
echo "</table>\n";
```

24 / 65

JDBC

- ▶ **JDBC**: Java DataBase Connectivity
- ▶ ODBC ile aynı mimari kavramları
 - ▶ değişik sürücü tipleri
- ▶ bağlantı için JDBC URL adresi
 - ▶ jdbc:<altprotokol>:<parametreler>
- ▶ Java ile SQL veri tiplerinin eşleştirilmesi

25 / 65

JDBC Sürücüleri

1. *Tip I*: köprüler
 - ▶ başka sistemlerin çağrılarını dönüştür (örneğin ODBC)
2. *Tip II*: Java olmayan bir sürücüyle doğrudan çeviri
 - ▶ veri kaynağının uygulama arayüzüne çevir (örneğin C++)
3. *Tip III*: ağ köprüleri
 - ▶ veri kaynağının uygulama arayüzüne çevirmesi için orta katman yazılımına bağlan
4. *Tip IV*: Java sürücüsüyle doğrudan çeviri
 - ▶ VTYS ile Java soketleri üzerinden iletişim kur

26 / 65

JDBC Akışı

- ▶ bir bağlantı nesnesi al
 - ▶ statik: DriverManager.getConnection() → Connection
- ▶ bağlantı üzerinde bir komut nesnesi yarat
 - ▶ Connection.createStatement() → Statement
- ▶ sorguyu çalıştır
 - ▶ okuma: Statement.executeQuery(query) → ResultSet
 - ▶ ekleme, güncelleme, silme: Statement.executeUpdate(query)
- ▶ sonuçları işle
- ▶ işi biten kaynakları kapat (sonuç kümesi, komut, bağlantı)

27 / 65

Sonuçların İşlenmesi

- ▶ ResultSet bir yineleyici
 - ▶ başta satır olup olmadığı: ResultSet.hasNext()
 - ▶ sıradaki satıra geç: ResultSet.next()
- ▶ satırdaki veriyi uygun tiplere dönüştürerek değişkenlere aktar
 - ▶ sütunun adıyla: ResultSet.getXXX(name)
 - ▶ sütunun sıra numarasıyla: ResultSet.getXXX(order)

28 / 65

Veri Tipi Dönüşümleri

SQL tipi	Java sınıfı	ResultSet metodu
BIT	Boolean	getBoolean()
CHAR	String	getString()
VARCHAR	String	getString()
DOUBLE	Double	getDouble()
FLOAT	Float	getDouble()
INTEGER	Integer	getInt()
REAL	Double	getFloat()
DATE	java.sql.Date	getDate()
TIME	java.sql.Time	getTime()
TIMESTAMP	java.sql.TimeStamp	getTimestamp()

29 / 65

JDBC Örneği

Örnek (veri tabanı sürücüsünün yüklenmesi)

```
try {
    Class.forName("org.postgresql.Driver");
} catch (ClassNotFoundException e) {
    // PostgreSQL driver not installed
}
```

30 / 65

JDBC Örneği

Örnek (veri tabanına bağlantı)

```
try {
    Connection conn = DriverManager.getConnection(
        "jdbc:postgresql:imdb", "itucs", "itucs"
    );
} catch (SQLException e) {
    // connection error
}
```

31 / 65

JDBC Örneği

Örnek (ekleme)

```
String query = "INSERT INTO MOVIE (TITLE, YR)"
    + " VALUES ('Casablanca', 1942)";
Statement stmt = conn.createStatement();
stmt.executeUpdate(query);
stmt.close();
```

32 / 65

Hazır Komutlar

- ▶ hazır komutlar parametre değerleri değiştirilerek yeniden kullanılabilir
- ▶ yaratma:
Connection.prepareStatement(query) → PreparedStatement
- ▶ sorguda parametreler için yer tutucu: '?'
 - ▶ çalıştırmadan önce değerleri ayarlanmalı:
PreparedStatement.setXXX(order, value)
- ▶ çalışma: PreparedStatement.executeQuery()
ya da PreparedStatement.executeUpdate()

33 / 65

Hazır Komut Örneği

Örnek (ekleme - hazır komutla)

```
String query = "INSERT INTO MOVIE (TITLE, YR)"
    + " VALUES (?, ?)";
PreparedStatement stmt =
    conn.prepareStatement(query);
for (Movie movie : getMovies()) {
    stmt.setString(1, movie.getTitle());
    stmt.setInt(2, movie.getYear());
    stmt.executeUpdate();
}
stmt.close();
```

34 / 65

Sonuç Kümesi Örneği

Örnek (bir yıldaki film başlıklarını listeleme)

```
String query = String.format(
    "SELECT TITLE FROM MOVIE WHERE (YR = %d)",
    year);
Statement stmt = conn.createStatement();
ResultSet results = stmt.executeQuery(query);
while (results.next()) {
    String title = results.getString("TITLE");
    System.out.println("Title: " + title);
}
results.close();
stmt.close();
```

35 / 65

JDBC Örneği

Örnek (silme)

```
String query = "DELETE FROM MOVIE" +
    " WHERE (ID = ?)";
PreparedStatement stmt =
    conn.prepareStatement(query);
stmt.setInt(1, movie.getId());
stmt.executeUpdate();
stmt.close();
```

36 / 65

JDBC Örneği

Örnek (güncelleme)

```
String query = "UPDATE MOVIE SET YR = ?" +  
    " WHERE (ID = ?)";  
PreparedStatement stmt =  
    conn.prepareStatement(query);  
stmt.setInt(1, movie.getYear());  
stmt.setInt(2, movie.getId());  
stmt.executeUpdate();  
stmt.close();
```

37 / 65

Kaynakları Kapama

- ▶ sonuç kümesi, komut gibi kaynakların `try - catch - finally` bloklarının `finally` kısmında kapatılması önerilir

38 / 65

Kaynakları Kapatma Örneği

```
Statement stmt = conn.createStatement();  
ResultSet results = null;  
try {  
    results = stmt.executeQuery(query);  
    ...  
} catch (SQLException e) {  
    ...  
} finally {  
    results.close();  
    stmt.close();  
}
```

39 / 65

Otomatik Kimlik Değerleri

- ▶ otomatik üretilen kimlik değerleri sorulmak isteniyorsa, komutu yaratırken `RETURN_GENERATED_KEYS` bayrağı kullanılır
- ▶ komutu çalıştırdıktan sonra:
`Statement.getGeneratedKeys() → ResultSet`

40 / 65

Otomatik Kimlik Değerleri Örneği

```
String query = "INSERT INTO ... ";  
PreparedStatement stmt =  
    connection.prepareStatement(query,  
        Statement.RETURN_GENERATED_KEYS);  
stmt.executeUpdate();  
  
ResultSet ids = stmt.getGeneratedKeys();  
// assuming there is one and only one result  
ids.next();  
int id = ids.getInt(1);
```

41 / 65

Toplu Komut Çalıştırma

- ▶ komutları biriktirerek topluca çalıştırmak hız kazandırabilir
- ▶ bağlantı "auto commit" kipinden çıkarılır
- ▶ biriktirme: `Statement.addBatch(query)`
- ▶ birikenleri çalıştırma: `Statement.executeBatch()`
- ▶ yazmayı sonlandırma: `Connection.commit()`

42 / 65

Toplu Komut Çalıştırma Örneği

```
stmt = conn.createStatement( ... );
conn.setAutoCommit(false);
int queryCount = 0;
int batchSize = 100;
for ( ... ) {
    stmt.addBatch(query);
    queryCount++;
    if (queryCount % batchSize == 0) {
        stmt.executeBatch();
        conn.commit();
    }
}
```

43 / 65

Önceden Okuma Miktarı

- ▶ JDBC sürücüler sorgudaki satırları önceden okur
- ▶ başarıyı artırır ama bellek harcamasını da artırır
- ▶ küçük bir önceden okuma miktarı bellek harcamasını azaltır: `Statement.setFetchSize(count)`

44 / 65

Önceden Okuma Miktarı Örneği

```
stmt.setFetchSize(1);

// MySQL
stmt.setFetchSize(Integer.MIN_VALUE)
```

45 / 65

Kayıtlı Yordamlar

- ▶ uygulamanın bazı işlevleri veri tabanı sunucusunda gerçekleştirilir
 - ▶ diller: SQL, PL/SQL, C, ...
 - ▶ **tavsiye edilmez**
 - ▶ taşınabilir değil
 - ▶ ölçeklenebilir değil
 - ▶ veri tabanı sunucuları iş mantığına göre optimize edilmiyor
- iş mantığı uygulama sunucusunda gerçekleştirilmeli

46 / 65

Fonksiyon Yaratma

Komut

```
CREATE FUNCTION
    function_name([parameter_type [, ...]])
    RETURNS return_type
    AS function_body
    LANGUAGE language_name
```

- ▶ birinci parametre \$1, ikinci parametre \$2, ...

47 / 65

SQL Fonksiyonu Örneği

Örnek (yeni puan hesaplanması)

\$1: eski puan, \$2: eski oy sayısı, \$3: yeni oy

```
CREATE FUNCTION NEW_SCORE(float, int, int)
    RETURNS float
    AS 'SELECT ($1*$2+$3) / ($2+1);'
    LANGUAGE 'sql'
```

48 / 65

Tetikler

Tanım

tetik: belirli olaylarda kendiliğinden etkinleştirilen fonksiyon

- bütünlüğü sağlamaya yardımcı olabilir

49 / 65

Tetik Yaratma

Komut

```
CREATE TRIGGER trigger_name
{ BEFORE | AFTER } { event [ OR ... ] }
ON table_name
[ FOR [ EACH ] { ROW | STATEMENT } ]
EXECUTE PROCEDURE function_name(...)
```

- PL/pgSQL:
 - old: çoklunun işlemden önceki değeri
 - new: çoklunun işlemden sonraki değeri

50 / 65

Tetik Örneği

Örnek (SCORE * VOTES değeri POINTS sütununda tutulsun)

```
CREATE FUNCTION UPDATE_MOVIE_POINTS()
RETURNS opaque
AS 'BEGIN
    new.POINTS = new.SCORE * new.VOTES;
    RETURN new;
END;'
LANGUAGE 'plpgsql'
```

51 / 65

Tetik Örneği

Örnek (POINTS sütunu güncellemelerde kendiliğinden hesaplınsın)

```
CREATE TRIGGER UPDATE_MOVIE
BEFORE INSERT OR UPDATE ON MOVIE
FOR EACH ROW
EXECUTE PROCEDURE UPDATE_MOVIE_POINTS()
```

52 / 65

Görüntüler

- türetilmiş tabloyu taban tablo gibi göstermek
- veri tabanı yapısındaki değişikliklerden kullanıcıların ve uygulama programlarının etkilenmemesi

53 / 65

Görüntü Yaratma

Komut

```
CREATE VIEW view_name AS
SELECT ...
```

- görüntü üzerindeki her işlemde
SELECT komutu yeniden çalıştırılır

54 / 65

Görüntü Örneği

Örnek

```
CREATE VIEW NEW_MOVIE AS
  SELECT ID, TITLE, YR FROM MOVIE
    WHERE (YR > 1995)

SELECT * FROM NEW_MOVIE
```

55 / 65

Görüntüde Güncelleme

- güncellemeler taban tablolarda yapılmalı
 - kural belirtilmeli

Kural Yaratma

```
CREATE RULE rule_name AS
  ON event TO view_name
  [ WHERE condition ]
  DO [ INSTEAD ] sql_statement
```

56 / 65

Görüntü Kuralı Örneği

Örnek

```
UPDATE NEW_MOVIE SET TITLE = '...'
  WHERE (ID = 1)

CREATE RULE UPDATE_TITLE AS
  ON UPDATE TO NEW_MOVIES
  DO INSTEAD
    UPDATE MOVIE SET TITLE = new.TITLE
      WHERE (ID = old.ID)
```

57 / 65

İzinler

- **özne:** kullanıcı, kullanıcı grubu
- **nesne:** tablo, sütun, görüntü, veri tabanı, ...
- nesnenin sahibi olan özne diğer öznelerin erişim yetkilerini belirler

58 / 65

SQL İzinleri

İzin Verme

```
GRANT permission_name [, ...]
  ON object_name TO subject_name
  [ WITH GRANT OPTION ]
```

İzin Kaldırma

```
REVOKE permission_name
  ON object_name FROM subject_name
```

59 / 65

İzin Örnekleri

Örnek (tablo için izin verme)

```
GRANT SELECT, INSERT, UPDATE ON MOVIE
  TO 'itucs'
```

Örnek (tablo için izin kaldırma)

```
REVOKE INSERT ON MOVIE
  FROM 'itucs'
```

60 / 65

Dizinler

- ▶ bazı işlemler sıralama gerektirir:
`ORDER BY`, `DISTINCT`, `GROUP BY`, `UNION`, ...
- ▶ dizinler aramaları hızlandırır
 - ▶ ekleme ve güncellemeleri yavaşlatır

Komut

```
CREATE [ UNIQUE ] INDEX index_name  
ON table_name(column_name [, ...])
```

61 / 65

Toplu Veri Aktarımı

- ▶ satırları tek tek eklemek ya da silmek yerine
ürüne özel toplu veri aktarımı komutları kullanılmalı
- ▶ tab ile ayrılmış değer dosyaları ile okuma-yazma

62 / 65

Toplu Veri Aktarımı

PostgreSQL

```
COPY table_name (column_name [, ...])  
TO 'output_file_path'
```

```
COPY table_name (column_name [, ...])  
FROM 'input_file_path'
```

63 / 65

Toplu Veri Aktarımı

MySQL

```
SELECT column_name [, ...] FROM table_name  
INTO OUTFILE 'output_file_path'
```

```
LOAD DATA INFILE 'input_file_path'  
INTO TABLE table_name (column_name [, ...])
```

64 / 65

Kaynaklar

Okunacak: Date

- ▶ Chapter 4: An Introduction to SQL
 - ▶ 4.6. Embedded SQL
- ▶ Chapter 9: Integrity
 - ▶ 9.11. Triggers (a Digression)
- ▶ Chapter 10: Views

Yardımcı Kaynak: Ramakrishnan, Gehrke

- ▶ Chapter 6: Database Application Development

65 / 65