

İstanbul Üniversitesi-Cerrahpaşa  
Bilgisayar Müh. Bölümü  
BIMU3064/CSBM3098 Veritabanı Yönetim Sistemleri

## I. DERS TANIMI

**Ders kodu:** BIMU3064/CSBM3098

**Ders adı:** Veritabanı Yönetim sistemleri

**Eposta grubu:** [veritabani2022@googlegroups.com](mailto:veritabani2022@googlegroups.com) (<http://groups.google.com/>)

Veritabanları (Ödev ve uygulama için indirip bilgisayarınıza kurmanız gerekmektedir):

- MySQL (MariaDB):
  - Tek başına: <https://dev.mysql.com/downloads/mysql/>
  - PHP+MySQL+Apache: <https://www.apachefriends.org/download.html>
- PostgreSQL: <https://www.postgresql.org/download/windows/>
- MongoDB: <https://www.mongodb.com/try/download/community>
- Firebase: <https://firebase.google.com/> (Sadece cloud versiyonu var)

**İçerik:** Bu derste veritabanı yönetim sistemlerine giriş yapılarak temel konular kapsanacaktır. İlişkisel veri modeli, E-R modeli veritabanı tasarımı, SQL, ileri SQL, tetikleyiciler (trigger), saklı yordamlar (stored procedures), veritabanı programlama (ODBC/JDBC) vb. konular işlenecektir.

**Ders hocası:** Atakan Kurt, [atakan.kurt@istanbul.edu.tr](mailto:atakan.kurt@istanbul.edu.tr) (212) 473 70 70 dahili 17450

**Ders asistanı:** Araş. Gör. Ramiz GÖRKEM, [ramiz.birdal@istanbul.edu.tr](mailto:ramiz.birdal@istanbul.edu.tr) (212) 473 70 70 dahili 17456

### **Ödevler:**

1. Ödevler pratik yaparak veritabanını öğrenmek için size verilmektedir. Ödevi yaptığınız takdirde iş girerken “Şu ve şu veritabanı sistemlerinde şunları yaptım” diyebilirsiniz. CVnizde ödevlerden bilgi birikimi/tecrübe olarak bahsedebilirsiniz.
2. Ödevi yaparken karşılaştığınız hataları google da aratırsanız çözümlerine hemen ulaşabilirsiniz. Çözümeyemediğin problemleri ders asistanına ofiste yada eposta ile sorabilirsiniz. Anlayamadığınız soruları bana eposta ile ulaşabilirsiniz. İnternette ve asistandan çözüm bulamadığınız problemleri benimle paylaşabilirsiniz.
3. Ders için oluşturulmuş eposta grubuna üye olunuz. Bana sorulan genel soruların cevaplarını grupla paylaşıyorum.
4. Ödevi uğraşarak, analayarak yapanlar sınavda da başarılı olacaklardır.
5. Ödevlerde kapak kullanmayınız. Mümkünse kağıtların arka yüzünü de kullanın. Her soruyu bir sayfaya yapıp sayfalarda boşlukla bırakmayınız. Ödev kağıtlarını bir zımba ile zımbalayarak verebilirsiniz. Ayrıca poşet dosyaya koymaya gerek yoktur.
6. Ödev kağıtlarını aldığınızda kendi cevaplarınızı web sitesinde yayınladığım doğru cevaplarla karşılaştırın. Eğer hala eksik/yanlış notlandırıldığını düşündüğünüz noktalar varsa önce asistana başvurunuz. Asistanla fikir birliğine varamazsanız bana da başvurabilirsiniz.

### **Ders kitabı:**

1. *Database Systems Concepts*, Abraham Silberschatz, et al, McGrawHill, 0-07-710365-3, <http://www.bell-labs.com/topic/books/db-book>, <http://www.mhhe.com/silberschatz>
2. Veri Tabanı Sistemleri, Dr. Yalçın Özkan, Alfa Yayınları, 2003, 975-297-360-4

## II. DERS İÇERİĞİ

- An introduction to database systems
- Entity-Relationship model
- Relational Data Model
- SQL
  - Data Definition Language
  - Data Manipulation Language
  - Data Control Language
  - Odbc/jdbc
- Integrity and Security
- Stored Procedures and Triggers
- NoSQL veritabanları
- Firebase veritabanı
- Indexing

### **III. NOTLANDIRMA**

5 Ödev (~30 puan) arasınnav (~20 points), 1 final (50 points).

### **IV Önemli Notlar**

- Devam zorunluluğu yoktur. Fakat dersi geçmeniz için derse devam etmeniz elzemdir.
- Derse geç kalmayınız.
- Derste bilgisayar, elektronik alet, cep telefonu kullanımı yasaktır. Ders başlamadan önce kapatınız.
- Ödevlerde/projelerde/sınavlarda kopya çekilmesi akademik suçtur ve disiplin işlemi yapılır.
- Sınıfta sessiz olunuz. Hoca sınıfa girdiğinde ders başlar. Derste sadece ders ile ilgileniniz.

Derste Kapsanacak Konular	
SQL	
<ul style="list-style-type: none"> <li>• Sample Databases               <ul style="list-style-type: none"> <li>○ School database</li> <li>○ Song database</li> <li>○ Crime database</li> <li>○ Soccer Database</li> <li>○ Bank Database</li> <li>○ Hospital Database</li> </ul> </li> <li>• RA-to-SQL conversion</li> <li>• Single table queries               <ul style="list-style-type: none"> <li>○ Duplicate removal (distinct)</li> <li>○ Ordering of results</li> <li>○ Column and table aliases</li> </ul> </li> <li>• Multi-table queries (joins)               <ul style="list-style-type: none"> <li>○ Inner joins</li> <li>○ Natural joins</li> <li>○ Outer joins</li> <li>○ Queries with multi-copies of a table</li> </ul> </li> <li>• NULLs</li> <li>• Set operations</li> </ul>	

<ul style="list-style-type: none"> <li>○ Union</li> <li>○ Intersect</li> <li>○ Except</li> <li>• Aggregate Functions <ul style="list-style-type: none"> <li>○ functions</li> <li>○ GROUP BY</li> <li>○ HAVING</li> <li>○ NULLs in aggregates</li> </ul> </li> <li>• Subqueries <ul style="list-style-type: none"> <li>○ Set membership test (value IN subquery)</li> <li>○ Set comparison (value &lt; ALL SOME subquery)</li> <li>○ Empty set test (NOT EMPTY (subquery))</li> <li>○ Uniqueness test of a set ( UNIQUE (subquery))</li> </ul> </li> <li>• Division queries</li> <li>• Query rewriting with alternative operators and subqueries <ul style="list-style-type: none"> <li>○ JOINing 2 tables in a single query</li> <li>○ Using IN and Subquery</li> <li>○ Using EXISTS and subquery</li> </ul> </li> <li>• Database Modification <ul style="list-style-type: none"> <li>○ Insert</li> <li>○ Update</li> <li>○ delete</li> </ul> </li> </ul>	
<p>Integrity Constraints (IC)</p> <ul style="list-style-type: none"> <li>• What is an IC</li> <li>• How is an IC is defined (CREATE/ALTER TABLE command)</li> <li>• How/when is an IC enforced (use a DB Modification example to get an IC error violation for all types of ICs)</li> <li>• Domain IC</li> <li>• Primary key IC</li> <li>• Unique key IC</li> <li>• Foreign key IC</li> <li>• NOT NULL IC</li> <li>• Single column IC (CHECK)</li> <li>• Multi-column (Table) IC (CHECK)</li> <li>• Assertions (multi-records and/or multi-table IC)</li> </ul>	
<p>Authorization</p> <ul style="list-style-type: none"> <li>• Users: Creation, dropping</li> <li>• Priviledges (on instances and schemas)</li> <li>• Granting permissions</li> <li>• Revoking permissions</li> <li>• Roles</li> </ul>	
<p>Database Programming: JDBC</p> <ul style="list-style-type: none"> <li>• Definition, necessity for JDBC</li> <li>• Database Programming model <ul style="list-style-type: none"> <li>○ Load the driver</li> </ul> </li> </ul>	

<ul style="list-style-type: none"> <li>○ Create a connection object</li> <li>○ Create a statement object</li> <li>○ Prepare and Execute a query, get results in a Resultset object</li> <li>○ Retrieve records in Resultset using a loop</li> <li>● PostgreSQL/MySQL/Oracle JDBC Examples</li> </ul>	
<p>Stored Procedures &amp; Functions</p> <ul style="list-style-type: none"> <li>● Definition</li> <li>● Necessity for stored pro &amp; funct</li> <li>● Advantages of stored pro &amp; funct</li> <li>● How to create, modify, drop stored pro &amp; funct</li> <li>● How/where to invoke stored pro &amp; funct</li> <li>● Difference between pro &amp; funct</li> <li>● Examples</li> </ul>	
<p>Entity &amp; Relationship Model</p> <ul style="list-style-type: none"> <li>● Entity Sets</li> <li>● Relationship Sets</li> <li>● Attributes</li> <li>● Mapping Cardinality constraint</li> <li>● Completeness constraints</li> <li>● Generalization</li> <li>● E-R to Relational Model</li> </ul>	
<p>Relational Data Model</p> <ul style="list-style-type: none"> <li>● Relational Algebra <ul style="list-style-type: none"> <li>○ Basic Operators</li> <li>○ Extended Operators</li> <li>○ Aggregate functions</li> </ul> </li> <li>● Example Queries</li> <li>● E-R Model to Relational Model Conversion</li> </ul>	