VERİ TABANI DERS NOTLARI

Ders #3: Relational Model (İlişkisel Model)

References

- Examples from *"Elmasri, Navathe*, Fund. of Database Systems, Addison Wesley"

Konu Başlıkları

- İlişkisel Veri Modeli (Relational Model)
- İlişkisel Model'de Kısıtlamalar (Constraints)
- İlişkisel Model'de Şema (Schemas) ve Olgu
- Dönüştürme: Varlık-Bağıntı Modeli (ER) → İlişkisel Model (RM)
  - **Varlık Kümelerini dönüştürme**
  - **Bağıntı Kümelerini dönüştürme**

Genel Bakış:

- Relational model: after hierarchical & network models
  - First commercial implementations available in early 1980s
  - Implemented in a large number of commercial system
- İlişkisel Model, bildirim (declarative) esaslı bir modeldir.
  - Bildirim esaslı model, kullanıcıya «NE İSTEDİĞİNİ» aktarma imkanı sağlar. Verinin nasıl yerleşeceği veya nasıl erişileceği ile ilgilenilmez. Diğer bir ifade ile, donanım ve gerçekleştirim ortamından veri bağımsızlığı *(data independence)* sağlanmış olur.
- İlişkisel model, SQL isimli veri işleme (tanımlama / sorgulama) dili kullanır.
- Model, geliştiricisi **Codd** (IBM Research): "**A Relational Model of Data for Large Shared Data Banks**," Communications of the ACM, June 1970. Yazarına **ACM Turing Award** ödülünü kazandırır (1981)
- Öncü ilişkisel VTYS örnekleri: System R (IBM) ve Ingres (UC-Berkeley)

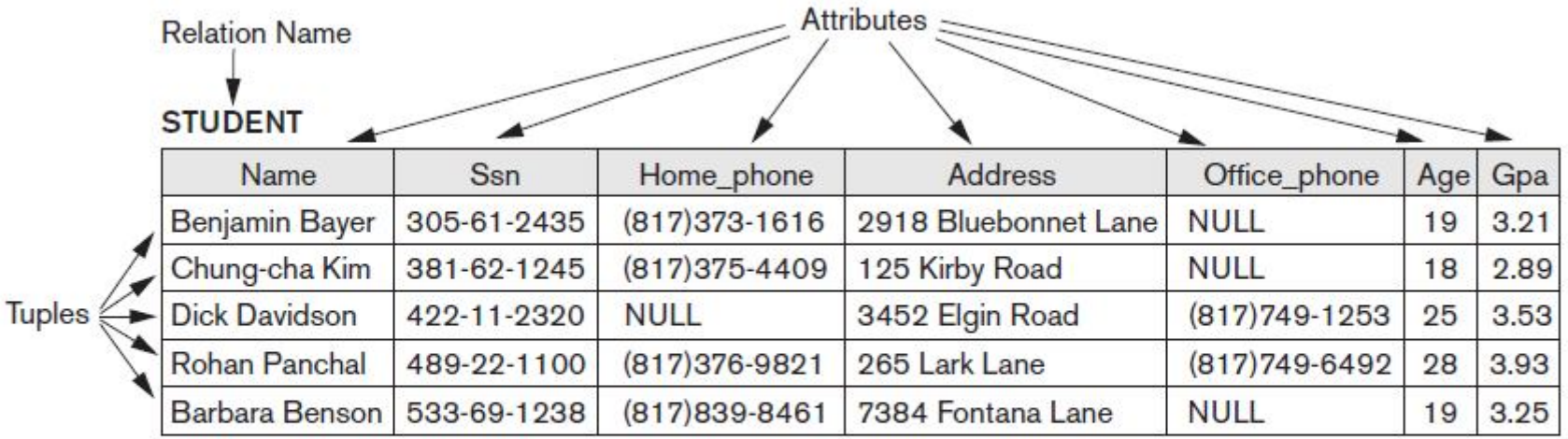
Relational Model Concepts

- İlişkisel veri modeli, VT'nındaki bütün «veri ve bağıntıların» tablolar (çizelgeler) olarak gösterimine dayalıdır. Yani, bu modelin temel yapı taşı TABLO (ÇİZELGE)dir
- Represents data as a collection of **relations**
- **Table** of values
  - Row
    - Represents a collection of related data values
    - Fact typically corresponds to a real-world entity or relationship
    - *Tuple*
  - Table name and column names: Interpret the meaning of the values in each row *attribute*

İlişkisel Model (Relational Model, RM)

- RM verileri ilişkiler topluluğu olarak modeller.
- **İlişki (relation)** iki boyutlu bir değerler tablosu (**table** of values).
- İlişkide, satır kümesi var (**set of rows)**. Satırlar **tuples-çoklu** diye de adlandırılır.
- Satır (çoklu) tekrarına izin verilmez.
- Her satırdaki veri elemanları belirli gerçekleri (**nesne** yada **bağıntı**) gösterir (**entity** or **relationship).**
- Her sütun-kolon (**column**) bir niteliği gösterir. Kolon başlığı, o kolondaki verinin anlamını belirtir (nitelik-attribute).
  - Nitelik yalın değer içermeli yani çok-değerli niteliğe izin yok
  - Her nitelik bir tanım kümesi (domain, type) elemanlarından bir değer alır

RM Concepts: İlişki Örneği



Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25

Figure 3.1
The attributes and tuples of a relation STUDENT.

Domains, Attributes, Tuples, and Relations

- **Domain** D
  - Set of atomic values
- **Atomic**
  - Each value indivisible
- Specifying a domain
  - **Data type** specified for each domain

Domains,Attributes,Tuples,Relations (cont'd.)

- **Relation schema** R
  - Denoted by $R(A_1, A_2, ..., A_n)$
  - Made up of a relation name R and a list of attributes, A_1, A_2, ..., A_n
- Attribute A_i
  - Name of a role played by some domain D in the relation schema R
- **Degree** (or **arity**) of a relation
  - Number of attributes n of its relation schema, n-tuple relation
  - **İlişkinin derecesi:** ilişki şemasındaki nitelik sayısı

Schema-Şema

- Bir ilişki şeması (**schema**, description):
  - R(A1, A2, ...An) biçiminde gösterilir.
  - R ilişki adı (**name**), A1, A2, ..., An ise ilişkinin nitelikleri(**attributes)**
  - **İlişkinin derecesi:** ilişki şemasındaki nitelik sayısı
- **Örnek**:

  CUSTOMER (Cust-id, Cust-name, Address, Phone#)

  ilişki ismi: CUSTOMER, ilişki derecesi: 4

  Nitelikler: Cust-id, Cust-name, Address, Phone#.
- A relation is a set of tuples (rows). A tuple is an ordered set of values (enclosed in angled brackets '< … >'). Each value is derived from an appropriate domain.
  - A row in CUSTOMER relation is a 4-tuple and consists of four values:
  - <**632895**, "John Smith", **"101 Main St. Atlanta, GA 30332"**, "(404)894-2000">

Domains, Attributes, Tuples, and Relations (cont'd.)

- **Relation** (or **relation state**)
  - Set of ***n*-tuples** $r = \{t_1, t_2, ..., t_m\}$
  - Each n-tuple t
    - Ordered list of n values $t = <v_1, v_2, ..., v_n>$
    - Each value v_i, $1 \leq i \leq n$, is an element of dom(A_i) or is a special NULL value
- Relation (or relation state) $r(R)$
  - **Mathematical relation** of degree n on the domains dom(A_1), dom(A_2), ..., dom(A_n)
  - **Subset** of the **Cartesian product** of the domains that define R:
    - $r(R) \subseteq (\text{dom}(A_1) \times \text{dom}(A_2) \times ... \times \text{dom}(A_n))$

Domains, Attributes, Tuples, and Relations (cont'd.)

- **Current relation state**
  - Relation state at a given time
  - Reflects only the valid tuples that represent a particular state of the real world
- Attribute names
  - Indicate different **roles**, or interpretations, for the domain

Characteristics of Relations

- Ordering of tuples in a relation
  - Relation defined as a set of tuples
  - Elements have no order among them
- Ordering of values within a tuple
  - Order of attributes and values is not that important as long as correspondence b/w attributes&values maintained
- Alternative definition of a relation
  - Tuple considered as a set of (<attribute>, <value>) pairs
  - Each pair gives the value of the mapping from an attribute A_i to a value v_i from dom(A_i)
- Use the first definition of relation
  - Attributes and the values within tuples are ordered
  - Simpler notation

Characteristics of Relations (cont'd.)

Figure 3.2
The relation STUDENT from Figure 3.1 with a different order of tuples.

STUDENT

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21

Characteristics of Relations (cont'd.)

- Values and NULLs in tuples
  - Each value in a tuple is atomic
  - **Flat relational model**
    - Composite and multivalued attributes not allowed
    - **First normal form** assumption
  - Multivalued attributes
    - Must be represented by separate relations
  - Composite attributes
    - Represented only by simple component attributes in basic relational model

Characteristics of Relations (cont'd.)

- NULL values
  - Represent the values of attributes that may be unknown or may not apply to a tuple
  - Meanings for NULL values
    - *Value unknown*
    - *Value exists but is not available*
    - *Attribute does not apply to this tuple (also known as value undefined)*

Characteristics of Relations (cont'd.)

- Interpretation (meaning) of a relation
  - **Assertion**
    - Each tuple in the relation is a **fact** or a particular instance of the assertion
  - **Predicate**
    - Values in each tuple interpreted as values that satisfy predicate

Definitions – *example*

- Formally, given R(A1, A2, ..., An)
  - r(R) ⊂ dom (A1) X dom (A2) X ....X dom(An)
  - R(A1, A2, …, An) is the **schema** of the relation
  - r(R): a specific **state** (or "value" or "population") of relation R – this is a *set of tuples* (rows)
    - r(R) = {t1, t2, …, tm} where each ti is an n-tuple
    - ti = <v1, v2, …, vn> where each vj *element-of* dom(Aj)
- Let R(A1, A2) be a relation schema:
  - Let dom(A1) = {0,1}, Let dom(A2) = {a,b,c}
  - Then: dom(A1) X dom(A2) is all possible combinations:
    {<0,a> , <0,b> , <0,c>, <1,a>, <1,b>, <1,c> }
    The relation state r(R) ⊂ dom(A1) X dom(A2)
  - For example: r(R) could be {<0,a> , <0,b> , <1,c> }
    - this is one possible state (population or extension) r of the relation R, defined over A1 and A2.
    - It has three 2-tuples: <0,a> , <0,b> , <1,c>

Definitions: Relational Database Schema

- A set S of relation schemas that belong to the same database.
- S is the name of the whole **database schema**
- S = {R1, R2, ..., Rn}
- R1, R2, …, Rn are the names of the individual **relation schemas** within the database S

- *Example:* COMPANY db schema w/ 6 relation schemas:

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

DEPT_LOCATIONS

Dnumber	Dlocation

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

Essn	Pno	Hours

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

Figure 5.5
Schema diagram for the COMPANY relational database schema.

Definition Summary

Informal Terms (Pratik model)		Formal Terms (Biçimsel model)
Table		Relation
Column Header		Attribute
All possible Column Values		Domain
Row		Tuple
Table Definition		Schema of a Relation
Populated Table		State of the Relation

Relational Model Constraints

- Constraints
  - Restrictions on the actual values in a db state. (i.e., **conditions** that must hold on **all** valid relation states)
  - Derived from the rules in the miniworld that the database represents
- There are <u>three *main types*</u> of constraints in the relational model:
  - **Key** constraints (*anahtar kısıtı*)
  - **Entity integrity** constraints (*varlık bütünlük kısıtı*)
  - **Referential integrity** constraints (*ima bütünlük kısıtı*)
- Another implicit constraint is the **domain** constraint
  - Every value in a tuple must be from the *domain of its attribute* (or it could be **null**, if allowed for that attribute)
- Yet another constraint is **Semantic constraints..**

Domain Constraints

- Typically include:
  - Numeric data types for integers and real numbers
  - Characters
  - Booleans
  - Fixed-length strings
  - Variable-length strings
  - Date, time, timestamp
  - Money
  - Other special data types

Key Constraints: Key def'n

- Key Constraint: No two tuples can have the same combination of values for all their attributes.
- **Superkey SK:** No two distinct tuples in any state *r* of *R* can have the same value for SK
- **Key**
  - Superkey of R
  - Removing any attribute *A* from *K* leaves a set of attributes *K* that is not a superkey of *R* any more
- **Key satisfies two properties:**
  - Two distinct tuples in any state of relation cannot have identical values for (all) attributes in key
  - Minimal superkey: Cannot remove any attributes and still have uniqueness constraint in above condition hold

Key Constraints: Key def'n

- **Candidate key**
  - Relation schema may have more than one key
- **Primary key** of the relation
  - Designated among candidate keys
  - Underline attribute
- Other candidate keys are designated as **unique keys**

Key Constraints and Constraints on NULL Values (cont'd.)

CAR

License_number	Engine_serial_number	Make	Model	Year
Texas ABC-739	A69352	Ford	Mustang	02
Florida TVP-347	B43696	Oldsmobile	Cutlass	05
New York MPO-22	X83554	Oldsmobile	Delta	01
California 432-TFY	C43742	Mercedes	190-D	99
California RSK-629	Y82935	Toyota	Camry	04
Texas RSK-629	U028365	Jaguar	XJS	04

Figure 3.4
The CAR relation, with two candidate keys: License_number and Engine_serial_number.

Integrity, Referential Integrity, and Foreign Keys

- **Entity integrity constraint**
  - No primary key value can be NULL
- **Referential integrity constraint**
  - Specified between two relations
  - Maintains consistency among tuples in two relations

Integrity, Referential Integrity, and Foreign Keys (cont'd.)

- **Foreign key** rules:
  - The attributes in FK have the same domain(s) as the primary key attributes PK
  - Value of FK in a tuple t_1 of the current state $r_1(R_1)$ either occurs as a value of PK for some tuple t_2 in the current state $r_2(R_2)$ or is NULL

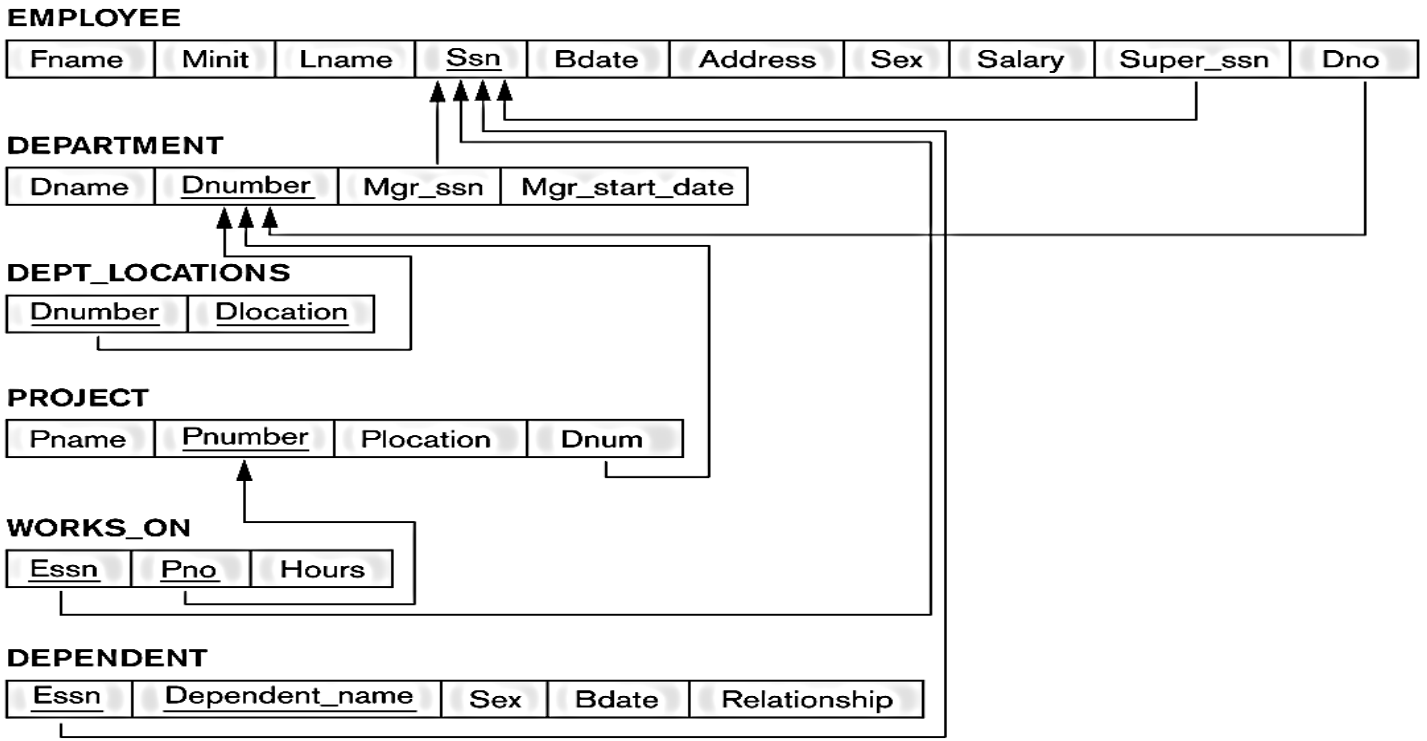
Integrity, Referential Integrity, and Foreign Keys (cont'd.)

- Diagrammatically display referential integrity constraints
  - Directed arc from each foreign key to the relation it references
- All integrity constraints should be specified on relational database schema

Displaying a relational database schema and its constraints

- Each relation schema can be displayed as a row of attribute names
- The name of the relation is written above the attribute names
- The primary key attribute (or attributes) will be underlined
- A foreign key (referential integrity) constraints is displayed as a directed arc (arrow) from the foreign key attributes to the referenced table
  - Can also point the the primary key of the referenced relation for clarity
- COMPANY **relational schema diagram:**

Figure 5.7
Referential integrity constraints displayed on the COMPANY relational database schema.



Update Operations and Dealing with Constraint Violations

- Each *relation* will have many tuples in its current relation state
- The *relational database state* is a union of all the individual relation states
- Whenever the database is changed, a new state arises
- Basic operations for changing the database:
  - INSERT a new tuple in a relation
  - DELETE an existing tuple from a relation
  - MODIFY an attribute of an existing tuple
- Next slide shows an example state for the COMPANY database

The Insert Operation

- Provides a list of attribute values for a new tuple t that is to be inserted into a relation R
- Can violate any of the four types of constraints
- If an insertion violates one or more constraints
  - Default option is to reject the insertion

The Delete Operation

- Can violate only referential integrity
  - If tuple being deleted is referenced by foreign keys from other tuples
  - **Restrict**
    - Reject the deletion
  - **Cascade**
    - Propagate the deletion by deleting tuples that reference the tuple that is being deleted
  - **Set null** or **set default**
    - Modify the referencing attribute values that cause the violation

The Update Operation

- Necessary to specify a condition on attributes of relation
  - Select the tuple (or tuples) to be modified
- If attribute not part of a primary key nor of a foreign key
  - Usually causes no problems
- Updating a primary/foreign key
  - Similar issues as with Insert/Delete

Populated database state for COMPANY

Figure 5.6
One possible database state for the COMPANY relational database schema.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

- INSERT a tuple.
- DELETE a tuple.
- MODIFY a tuple.
- Integrity constraints should not be violated by the update operations.
- Several update operations may have to be grouped together.
- Updates may **propagate** to cause other updates automatically. This may be necessary to maintain integrity constraints.

Specifying constraints in SQL

```
CREATE TABLE DEPT (
  DNAME   VARCHAR(10)  NOT NULL,
  DNUMBER INTEGER    NOT NULL,
  MGRSSN  CHAR(9) DEFAULT '000',
  MGRSTARTDATE      CHAR(9),
  PRIMARY KEY (DNUMBER),
  UNIQUE (DNAME),
  FOREIGN KEY (MGRSSN) REFERENCES EMP
                ON DELETE SET DEFAULT ON UPDATE CASCADE);
 CREATE TABLE EMP(
   ENAME    VARCHAR(30)NOT NULL,
   ESSN     CHAR(9),
   BDATE    DATE,
   DNO      INTEGER  DEFAULT 1,
   SUPERSSN CHAR(9),
   PRIMARY KEY (ESSN),
   FOREIGN KEY (DNO) REFERENCES   DEPT ON DELETE SET
   DEFAULT ON UPDATE  CASCADE,
   FOREIGN KEY (SUPERSSN) REFERENCES EMP ON DELETE SET
                NULL ON UPDATE CASCADE);
```

ÖNEMLİ: Yukarıdaki SQL DDL tablo oluşturma komutları 2 farklı sıra için de (DEPT, EMP ve EMP,DEPT) çalıştırmak mümkün olmuyor. Çünkü imalar mevcut omayan başka bir tabloya işaret edemez. O yüzden, ima kısıtlarını temel tabloyu oluşturduktan sonra ALTER TABLE ... yardımcı komutu ile ekleyebilriz.

Aynı durum DROP TABLE için de geçerli.

Populate DB with SQL

- Aynı dikkat, INSERT TABLE için de geçerli. Bu sefer UPDATE komutu yardımı ile eksikler tamamlanır. O yüzden önceki sayfadaki VT'da eklemelerin aşağıdaki gibi olması gerek: örneğin:

```
INSERT INTO EMP VALUES
  ('James', '888665555','10-NOV-27',null,null);
INSERT INTO EMP VALUES
  ('Franklin','T','Wong','333445555','08-DEC-45','638 Voss, Houston,
TX','M',40000,'888665555',null);
.....
INSERT INTO DEPT VALUES ('Research', 5, '333445555', '22-MAY-78');
INSERT INTO DEPT VALUES ('Headquarters', 1, '888665555', '19-JUN-71');

UPDATE employee SET dno = 5 WHERE ssn = '333445555';
UPDATE employee SET dno = 1 WHERE ssn = '888665555';
```

Exercise-1

(Taken from Exercise 5.15)

Consider the following relations for a database that keeps track of student enrollment in courses and the books adopted for each course: **Draw a relational schema diagram**

STUDENT(SSN, Name, Major, Bdate)

COURSE(Course#, Cname, Dept)

ENROLL(SSN, Course#, Quarter, Grade)

BOOK_ADOPTION(Course#, Quarter, Book_ISBN)

TEXT(Book_ISBN, Book_Title, Publisher, Author)