



Constraints

Integrity Constraints

- Constraints are **conditions** that specify restrictions on the database state
 - Protects database from errors
- Types of constraints captured in the relational model (i.e., they are automatically checked by DBMS):
 - **Domain** constraints
 - **Key** constraints
 - **Entity integrity** constraints
 - **Referential integrity** constraints
- Constraints not captured in the relational model
 - **semantic constraints** (business rules)
 - ◆ E.g., “Buy 5 Get 1 Free!”, “Can’t view profile unless you’ve logged in”

University Database

STUDENT	Id	Name	Address	Status
	111111111	John Doe	123 Main St.	Freshman
	666666666	Joseph Public	666 Hollow Rd.	Sophomore
	111223344	Mary Smith	1 Lake St.	Freshman
	987654321	Bart Simpson	Fox 5 TV	Senior
	023456789	Homer Simpson	Fox 5 TV	Senior
	123454321	Joe Blow	6 Yard Ct.	Junior

PROFESSOR	Id	Name	DeptId
	101202303	John Smyth	CS
	783432188	Adrian Jones	MGT
	121232343	David Jones	EE
	864297531	Qi Chen	MAT
	555666777	Mary Doe	CS
	009406321	Jacob Taylor	MGT
	900120450	Ann White	MAT

COURSE	CrsCode	DeptId	CrsName	Descr
	CS305	CS	Database Systems	On the road to high-paying job
	CS315	CS	Transaction Processing	Recover from your worst crashes
	MGT123	MGT	Market Analysis	Get rich quick
	EE101	EE	Electronic Circuits	Build your own computer
	MAT123	MAT	Algebra	The world where $2 * 2 \neq 4$

TRANSCRIPT	StudId	CrsCode	Semester	Grade
	666666666	MGT123	F1994	A
	666666666	EE101	S1991	B
	666666666	MAT123	F1997	B
	987654321	CS305	F1995	C
	987654321	MGT123	F1994	B
	123454321	CS315	S1997	A
	123454321	CS305	S1996	A
	123454321	MAT123	S1996	C
	023456789	EE101	F1995	B
	023456789	CS305	S1996	A
	111111111	EE101	F1997	A
	111111111	MAT123	F1997	B
	111111111	MGT123	F1997	B

TEACHING	ProfId	CrsCode	Semester
	009406321	MGT123	F1994
	121232343	EE101	S1991
	555666777	CS305	F1995
	864297531	MGT123	F1994
	101202303	CS315	S1997
	900120450	MAT123	S1996
	121232343	EE101	F1995
	101202303	CS305	S1996
	900120450	MAT123	F1997
	783432188	MGT123	F1997
	009406321	MGT123	F1997

Domain Constraint

- **Domain constraint**

- The value of each attribute in a tuple must be **atomic and consistent with its domain** (or it could be **null**, if allowed for that attribute)

STUDENT	Id	Name	Status	Address
	111223344	Mary Smith	Freshman	1 Lake St.
	987654321	Bart Simpson	Senior	Fox 5 TV
	111111111	John Doe	Freshman	123 Main St.
	023456789	Homer Simpson	Senior	Fox 5 TV
	666666666	Joseph Public	Sophomore	666 Hollow Rd.
	123454321	Joe Blow	Junior	6 Yard Ct.

Violations of Domain Constraints

STUDENT	Id	Name	Status	Address
	111223344	Mary Smith	Freshman	1 Lake St.
	987654321	Bart Simpson	Senior	Fox 5 TV
	111111111	John Doe	Freshman	123 Main St.
	023456789	Homer Simpson	Senior	Fox 5 TV
	666666666	Joseph Public	Sophomore	666 Hollow Rd.
	123454321	Joe Blow	Junior	6 Yard Ct.

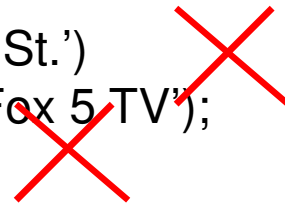
```
CREATE TABLE Student (  
    Id INT,  
    Name CHAR(30),  
    Status ENUM( 'Freshman', 'Sophomore', 'Junior', 'Senior'),  
    Address CHAR(30) );
```

Inserting the tuple (211223344, 'Billy Bob', 3, '46 Grand Ave.')

Inserting the tuple ('311211111', 'Mark Smith', 'Senior', '1 Oak St.')

Inserting the tuple (333333333, 'Bart Simpson', 'Graduate', 'Fox 5 TV');

OK



Key Constraint

- Every relation must have a key
- A key is a subset of attributes $\{A_1, \dots, A_m\}$ of a relation schema R with the following properties:
 - **Uniqueness:**
 - ◆ No two tuples in any valid relation state will have the same combination of values for A_1, \dots, A_m
 - **Minimality:**
 - ◆ No subset of A_1, \dots, A_m has the uniqueness property (no subset of a key can also be selected as a key)
- Sometimes row-ids or sequential numbers are assigned as keys to identify the rows in a table
 - Called *artificial key* or *surrogate key*

Key Constraint

- **Superkey** – a set of attributes containing a key
 - Satisfies uniqueness property (minimality property not necessary)
 - Every key is a superkey, but not vice-versa

CAR

<u>License_number</u>	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

- {LicenseNo} is a key
 - ◆ It is also a superkey of CAR
- {LicenseNo, Make} is a superkey but not a key

Key Constraint

- A relation can have several keys:
 - If a relation has several keys, they are referred to as **candidate keys** (one of them is designated as the **primary key**)
 - The primary key attributes are underlined in the relation schema
- Examples:
 - ◆ CAR(LicenseNo, EngineSerialNo, Make, Model, Year)
 - Candidate keys: {LicenseNo} or {EngineSerialNo}
 - Primary key: {LicenseNo}
 - ◆ STUDENT(Id, Name, Address, PhoneNo)
 - Candidate keys: {Name, Address} or {Id}
 - Primary key: {Id}

Exercise

Identify the candidate and primary keys for each of the following schema:

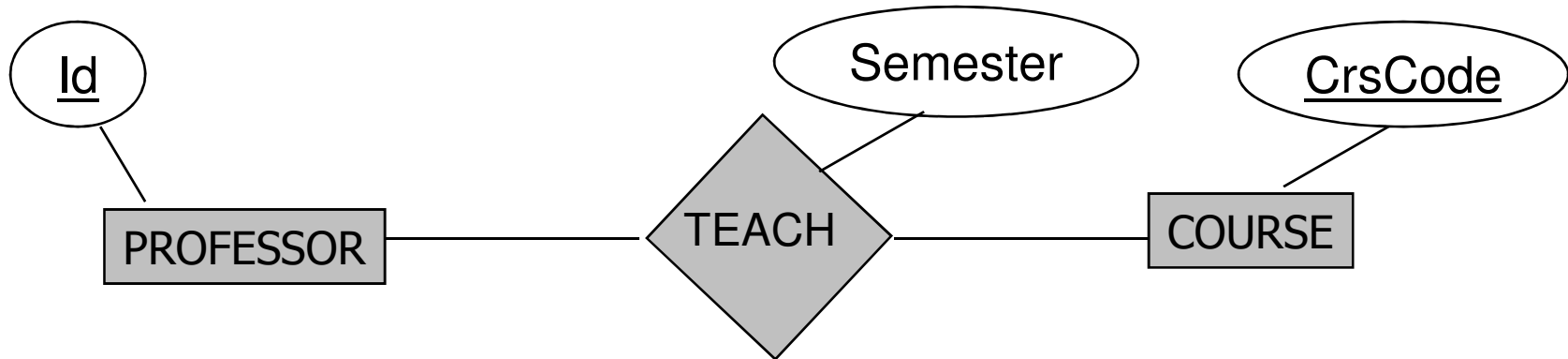
STUDENT (Id:INTEGER, Name:STRING, Address:STRING, Status:STRING)
PROFESSOR (Id:INTEGER, Name:STRING, DeptId:STRING)
COURSE (DeptId:STRING, CrsCode:STRING, CrsName:STRING, Descr:STRING)
TRANSCRIPT (StudId:INTEGER, CrsCode:STRING, Semester:STRING, Grade:STRING)
TEACHING (ProfId:INTEGER, CrsCode:STRING, Semester:STRING)
DEPARTMENT(Id: STRING, Name: STRING, Address: STRING)

Entity Integrity Constraint

- *Primary key attributes* PK cannot have null values in any tuple
 - This is because primary key values are used to *identify* the individual tuples.
 - If PK has several attributes, null is not allowed in any of these attributes
- Note: Other attributes of R can also be constrained to disallow null values, even though they are not members of the primary key.

```
CREATE TABLE Student (  
    Id INT PRIMARY KEY,  
    Name CHAR(30) NOT NULL,  
    Status ENUM( 'Freshman', 'Sophomore', 'Junior', 'Senior'),  
    Address CHAR(30) );
```

Relationship Constraints



PROFESSOR	Id	Name	DeptId
	101202303	John Smyth	CS
	783432188	Adrian Jones	MGT
	121232343	David Jones	EE
	864297531	Qi Chen	MAT
	555666777	Mary Doe	CS
	009406321	Jacob Taylor	MGT
	900120450	Ann White	MAT

TEACHING	ProfId	CrsCode	Semester
	009406321	MGT123	F1994
	121232343	EE101	S1991
	555666777	CS305	F1995
	864297531	MGT123	F1994
	101202303	CS315	S1997
	900120450	MAT123	S1996
	121232343	EE101	F1995
	101202303	CS305	S1996
	900120450	MAT123	F1997
	783432188	MGT123	F1997
	009406321	MGT123	F1997

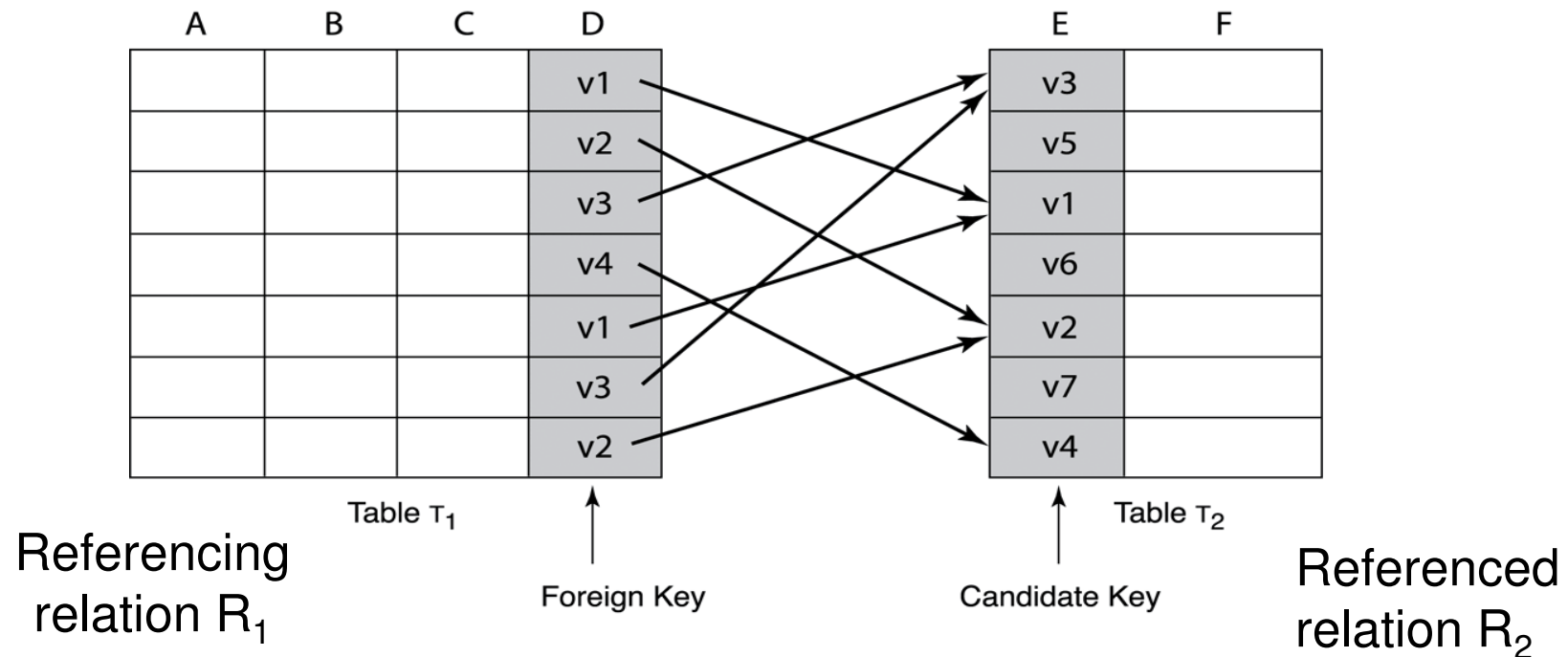
In relational databases, it is common for tuples in one relation to reference tuples in the same or other relations

Referential Integrity

- Referential integrity constraint
 - Tuples in the referencing relation must refer to tuples that exist in the referenced relation
- Referential integrity is violated if the referenced tuple does not exist in the appropriate relation
 - A tuple (123456789, MGT123, F1994) in TEACHING relation violates referential integrity constraint because there is no tuple with Id = 123456789 in PROFESSOR relation
- Referential integrity constraint is enforced in the DB using FOREIGN KEY on attributes in the referencing relation

Referential Integrity

- Tuples in the **referencing relation** R_1 have attributes FK (called **foreign key** attributes) that reference the **primary key**** attributes PK of the **referenced relation** R_2



** The book considers PK to be primary key, but in general, PK can be any candidate key

Referential Integrity

$R_1(\text{FK})$ references $R_2(\text{PK})$

- FK and PK are attributes that have the same domains (although not necessarily the same attribute names)
- The value in the FK of the **referencing relation** R_1 can be **either**:
 - An existing value of a corresponding PK in the **referenced relation** R_2 , or
 - **Null**
 - ◆ In this case, the FK **cannot** be part of the primary key for R_1
- R_1 and R_2 need not be distinct relations
 - Employee(Id:INT, MgrId:INT,)

Example

TEACHING(ProfId:INTEGER, CrsCode:STRING, Semester:STRING)

PROFESSOR(Id:INTEGER, Name:STRING, DeptId:STRING)

TEACHING (ProfId) REFERENCES PROFESSOR (Id)

- Not all rows in PROFESSOR need to be referenced
 - Some PROFESSORs may not have taught any classes yet
- Names of FK and PK attributes need not be the same
 - ProfId of Teaching references Id of Professor
- Value of a foreign key can be NULL
 - The PROFESSOR who teaches a course might not be known yet
 - Caution: ProfId cannot be part of the primary key for TEACHING

Exercise

- Identify all the referential integrity constraints in the student registration system

STUDENT(Id:INTEGER, Name:STRING, Address:STRING, Status:STRING)

PROFESSOR(Id:INTEGER, Name:STRING, DeptId:STRING)

COURSE(CrsCode:STRING, DeptId:STRING, CrsName:STRING, Descr:STRING)

TRANSCRIPT(StudId:INTEGER, CrsCode:STRING, Semester:STRING, Grade:STRING)

TEACHING(ProfId:INTEGER, CrsCode:STRING, Semester:STRING)

DEPARTMENT(Id:STRING, Name:STRING, Address:STRING)

Exercise

PROFESSOR(DeptId) REFERENCES DEPARTMENT(Id)

COURSE(DeptId) REFERENCES DEPARTMENT(Id)

TRANSCRIPT(StudId) REFERENCES STUDENT(Id)

TRANSCRIPT(CrsCode, Semester) REFERENCES
TEACHING(CrsCode, Semester)

TEACHING(ProfId) REFERENCES PROFESSOR(Id)

TEACHING(Crscod) REFERENCES COURSE(CrsCode)

TRANSCRIPT(CrsCode) REFERENCES COURSE(CrsCode) is
redundant since it is implied by Teaching(Crscod).

Relational Database Schema

- A set of relation schemas that belong to the same database
 - $S = \{R_1, R_2, \dots, R_n\}$
 - S is the name of the whole database schema
 - R_1, R_2, \dots, R_n are the names of individual relation schemas within the database S
- Referential integrity constraints are displayed as directed arcs from the foreign key attributes to the referenced table
 - Can also point to the primary key of the referenced relation for clarity

Example: COMPANY Database Schema

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

DEPT_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

WORKS_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

Figure 5.5
Schema diagram for
the COMPANY
relational database
schema.

Referential Integrity Constraints

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

DEPT_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

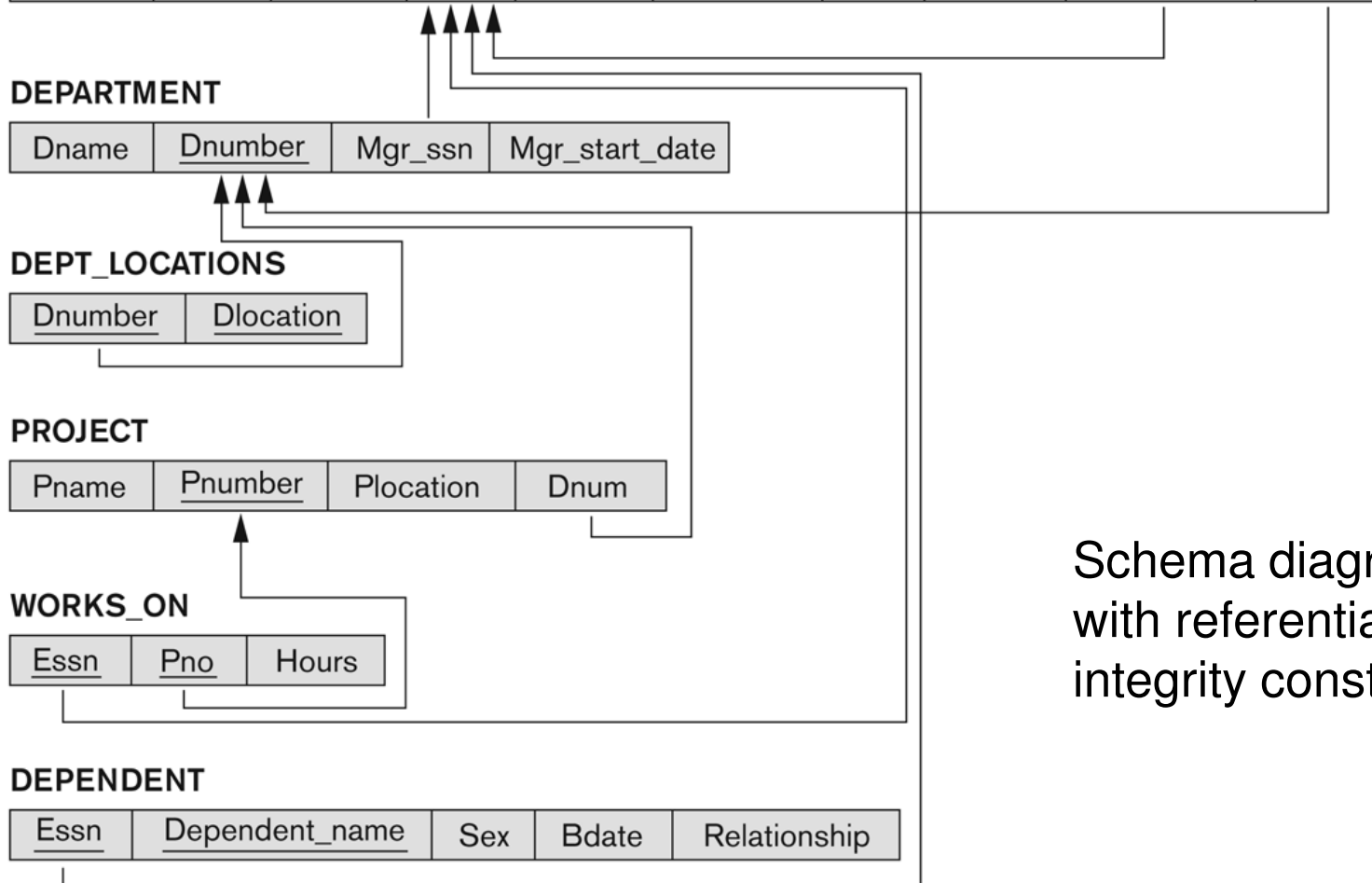
WORKS_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

Schema diagram
with referential
integrity constraints



Relational Database State

- Relational database state is the union of all the relation states

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	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	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	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

DEPT_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
1	Houston
4	Stafford
5	Bellaire
5	Sugarland

PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4

COMPANY Database

WORKS_ON

<u>Essn</u>	<u>Pno</u>	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

Update Operations on Relations

- Whenever the database is changed/updated, a new database state arises
- Basic operations for updating the database:
 - INSERT a new tuple into a relation
 - DELETE an existing tuple from a relation
 - UPDATE/MODIFY an attribute of an existing tuple in a relation
- Integrity constraints must not be violated by the update operations
 - Updates may propagate to cause other updates automatically. This may be necessary to maintain integrity constraints.

Example

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	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

WORKS_ON

<u>Essn</u>	<u>Pno</u>	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

If we change the SSN of John Smith to 111111111, this may require updates to some of the tuples in Works_ON relation (depending on the update option used)

Possible violations for INSERT

- Domain constraint
 - if one of the attribute values provided for the new tuple is not of the specified attribute domain
- Key constraint
 - if the value of a key attribute in the new tuple already exists in another tuple in the relation
- Entity integrity
 - if the primary key value is null in the new tuple
- Referential integrity
 - if a foreign key value in the new tuple references a primary key value that does not exist in the referenced relation

Example

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	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

- ◆ Insert <'John', 'M', 'Doe', **NULL**, '1977-01-01', '123 Main, TX', 'M', 45000, NULL, 4> into EMPLOYEE **will violate entity constraint**
- ◆ Insert <'Mary', 'M', 'Doe', **123456789**, '1977-01-01', '123 Main, TX', 'M', 45000, NULL, 4> into EMPLOYEE **will violate key constraint**
- ◆ Insert <'Tom', NULL, 'Doe', **'444444444'**, **1957**, NULL, 'M', **'100K'**, NULL, 4> into EMPLOYEE **will violate domain constraint**

Example

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	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

WORKS_ON

<u>Essn</u>	<u>Pno</u>	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

```
CREATE TABLE WORKS_ON (  
    Essn INT,  
    Pno INT,  
    Hours FLOAT,  
    PRIMARY KEY (Essn, Pno),  
    FOREIGN KEY Essn REFERENCES EMPLOYEE(Ssn)  
    FOREIGN KEY Pno REFERENCES PROJECT(Pnum));
```

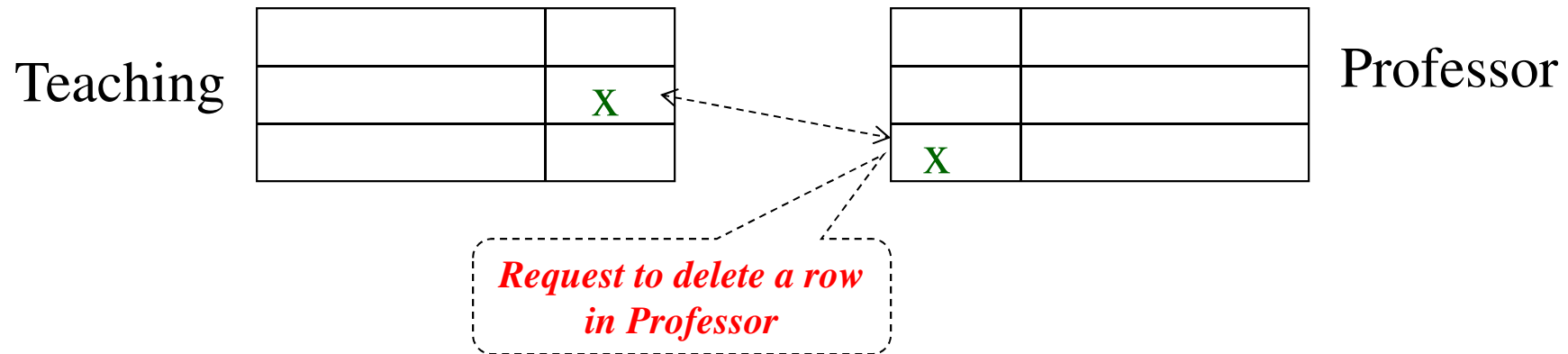
- ◆ Insert <999999999,1,34.5> into WORKS_ON **will violate referential integrity constraint**

Possible violations for DELETE

- DELETE may violate only referential integrity:
 - If the primary key value of the tuple being deleted is referenced by other tuples in the database
 - ◆ Can be remedied by several actions: RESTRICT, CASCADE, SET NULL
 - One of the above options must be specified for each foreign key constraint

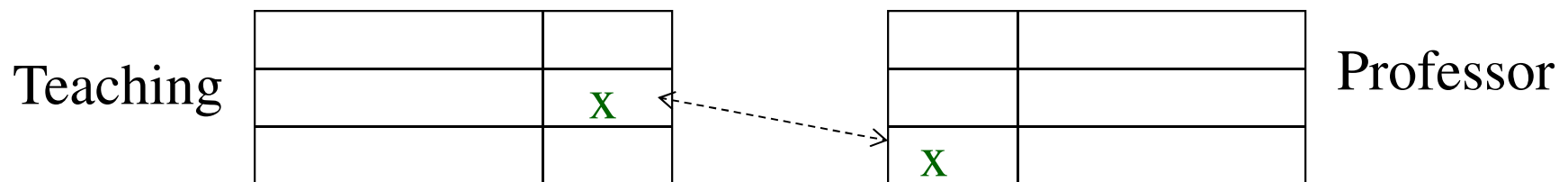
Example

- Deletion from referenced relation Professor:



- RESTRICT: Reject if row in Teaching references row to be deleted
- SET NULL: Set value of foreign key in Teaching to NULL
- SET DEFAULT: Set value of foreign key in Teaching to default value
- CASCADE: Delete referencing row(s) in Teaching as well

Example (in SQL)

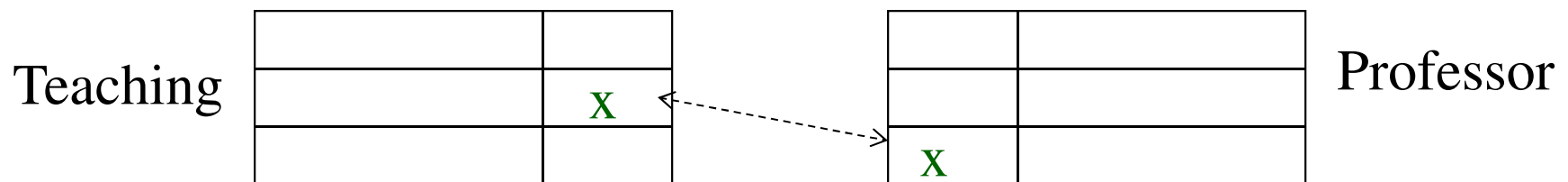


```
CREATE TABLE Teaching (  
  ProfId INT,  
  CrsCode CHAR(10),  
  Semester CHAR(10),  
  PRIMARY KEY (CrsCode, Semester),  
  FOREIGN KEY (ProfId) REFERENCES Professor(Id)  
    ON DELETE SET NULL,  
  FOREIGN KEY (CrsCode) REFERENCES COURSE(CrsCode)  
    ON DELETE RESTRICT  
);
```

Possible violations for UPDATE

- Domain, key, entity integrity, and referential integrity constraints may be violated, depending on the attribute being updated:
 - Updating the primary key (PK):
 - ◆ Similar to a DELETE operation followed by an INSERT operation
 - ◆ Need to specify similar options to DELETE
 - Updating a foreign key (FK):
 - ◆ May violate referential integrity
 - Updating an ordinary attribute (neither PK nor FK):
 - ◆ Can only violate domain constraints

Example (in SQL)



```
CREATE TABLE Teaching (  
  ProfId INT,  
  CrsCode CHAR(10),  
  Semester CHAR(10),  
  PRIMARY KEY (CrsCode, Semester),  
  FOREIGN KEY ProfId REFERENCES Professor(Id)  
    ON DELETE SET NULL  
    ON UPDATE SET NULL,  
  FOREIGN KEY CrsCode REFERENCES COURSE(CrsCode)  
    ON DELETE RESTRICT  
    ON UPDATE CASCADE  
);
```