Relational Data Model and Relational DB Constraints

406.426 Design & Analysis of Database Systems

Jonghun Park

jonghun@snu.ac.kr

Dept. of Industrial Engineering

Seoul National University

outline

- relational model concepts
- relational model constraints and relational database schemas
- update operations and dealing with constraint violations

history

- the relational model was first introduced by Ted Codd of IBM research in 1970
- based on **set theory** and **first-order predicate logic**
- first generation: Oracle, Ingress, SQL/DS
- current commercial systems: Oracle, DB2, Informix, SQL server

relational model concepts

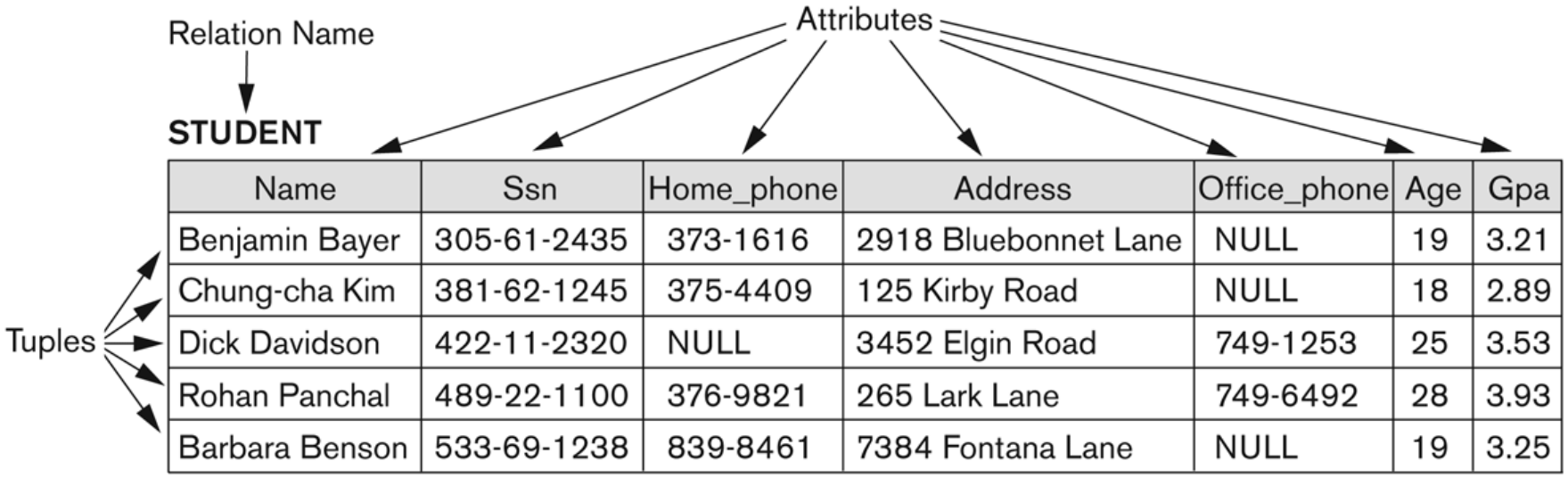
- the relational model represents the database as a **collection of relations**
  - each relation resembles a **table** of values
  - each **row** in the table represents a **collection of related data values**
  - all values in a **column** are of the **same data type**
- terminology
  - a row is called a **tuple**
  - a column header is called an **attribute**
  - the table is called a **relation**
  - the **data type** describing the types of values that can appear in each column is represented by a **domain**

domains, attributes, tuples, and relations

- a **domain** *D* is a set of **atomic values**
  - atomic: each value in the domain is **indivisible** as far as the relational model is concerned
  - specified by use of a **data type** from which the data values forming the domain are drawn
  - domain is given a **name**, **data type**, and **format**
  - e.g., GPA
- a **relation schema** (relational intension) $R(A_1, A_2, \ldots, A_n)$ is made up of a relation **name** *R* and a list of **attributes** $A_1, A_2, \ldots, A_n$
  - each attribute A_i is the **name of a role** played by some domain *D* in *R*
  - *D* is called the domain of A_i, and denoted by $dom(A_i)$
  - *n*: the degree of a relation
  - e.g., STUDENT (Name, SSN, Age, Phone, GPA)

domains, attributes, tuples, and relations (cont.)

- a **relation** (relation extension) $r(R)$ of the relation schema $R(A_1, A_2, \ldots, A_n)$ is a **set** of **n-tuples** $r = \{t_1, t_2, \ldots, t_m\}$
  - each n-tuple t is an **ordered list** of n values $t = <v_1, v_2, \ldots, v_n>$, where each value v_i, is an element of $dom(A_i)$ or is a special null value
  - the i-th value in tuple t, is referred to as $t[A_i]$
  - that is, $r(R) \subseteq (dom(A_1) \times \ldots \times dom(A_n))$



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

Figure 5.1
The attributes and tuples of a relation STUDENT.

characteristics of relations

- **tuple ordering is not part of a relation definition**
- however, the ordering of values in a tuple is important
- an alternative definition of a relation
  - each tuple t_i in $r = \{t_1, t_2, \ldots, t_m\}$ is a mapping from R to D, and $D = dom(A_1) \cup dom(A_2) \cup \ldots \cup dom(A_n)$
    - $t[A_i]$ must be in $dom(A_i)$ for $1 \leq i \leq n$ for each mapping t in r
  - a tuple can be considered as a **set of (<attribute>, <value>) pairs**
  - now the **ordering of attributes becomes unimportant**

STUDENT	Name	SSN	HomePhone	Address	OfficePhone	Age	GPA
	Dick Davidson	422-11-2320	null	3452 Elgin Road	749-1253	25	3.53
	Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	null	19	3.25
	Charles Cooper	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
	Katherine Ashly	381-62-1245	375-4409	125 Kirby Road	null	18	2.89
	Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	null	19	3.21

t = < (Name, Dick Davidson),(SSN, 422-11-2320),(HomePhone, null),(Address, 3452 Elgin Road),
(OfficePhone, 749-1253),(Age, 25),(GPA, 3.53)>

t = < (Address, 3452 Elgin Road),(Name, Dick Davidson),(SSN, 422-11-2320),(Age, 25),
(OfficePhone,749-1253),(GPA, 3.53),(HomePhone, null)>

characteristics of relations (cont.)

- null values have several meanings: “value unknown”, “not available”, “not applicable”, …
- the relation schema can be interpreted as a **declaration** or a type of assertion
  - some relations may represent facts about **entities**, whereas other relations may represent facts about **relationships**: STUDENT(Name, SSN, GPA) and MAJORS(SSN, Dept)

relational model constraints

- the state of the whole database will correspond to the **states of all its relations** at a particular point in time
- there are generally many constraints on the actual values in a database state
  - inherent **model-based** constraints: constraints that are inherent in the relational data model
    - e.g., a relation cannot have duplicate tuples
  - **schema-based** constraints: constraints that can be **directly expressed in the schemas** of the data model
    - domain constraints, constraints on nulls, key constraints, entity integrity constraints, referential integrity constraints
  - **application-based** constraints: expressed and enforced by the application programs

domain constraints and constraints on null values

- specify that within each tuple, the value of each attribute *A* must be an **atomic value** from *dom*(*A*)
  - example: characters, booleans, fixed-length strings, variable-length strings, date, time, …
- constraints on null values: specifies whether null values are or are not permitted

key constraints

- a relation is a **set** of tuples -> all elements of a set should be **distinct**
- **superkey** (SK) of the relation schema *R*
  - **subset of attributes** of *R* with the property that no two tuples in any relation state *r* of *R* should have the same combination of values for these attributes
  - i.e., $t_1[SK] \neq t_2[SK], \ \forall t_1, t_2$
  - specifies a **uniqueness** constraint that no two distinct tuples in any state *r* of *R* can have the same value for SK
  - every relation has at least one default superkey -> why?
- **key** (K) of *R* is a superkey of *R* with the additional property that removing any attribute *A* from *K* leaves a set of attributes *K'* that is not a superkey of *R* any more
  - hence, key is a **minimal superkey**
  - example: {SSN, Name, Age}, {SSN}

candidate keys

- a relation schema may have **more than one key** -> each of the keys is called a **candidate** key
- we can designate one of the candidate keys as the **primary key** of the relation

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 5.4
The CAR relation, with two candidate keys: License_number and Engine_serial_number.

relational database schemas

- a relational database usually contains many relations, with tuples in relations that are related in various ways
- a **relational database schema** S is a **set of relation schemas** $S = \{R_1, R_2, \ldots, R_m\}$ and a **set of integrity constraints** IC
- a **relational database state DB** of S is a set of relation states DB = $\{r_1, r_2, \ldots, r_m\}$ such that each r_i is a state of R_i and such that the r_i relation states satisfy the IC
- a database state that does not obey all the ICs is called an **invalid** state; valid state, o.w.

example of a relational database schema

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.

- attributes that represent the same real-world concept may or **may not have identical names** in different relations
- attributes that represent different concepts **may have the same name** in different relations

example of a database state

EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	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	1968-07-19	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

DEPT_LOCATIONS	DNUMBER	DLOCATION
	1	Houston
	4	Stafford
	5	Bellaire
	5	Sugarland
	5	Houston

DEPARTMENT	DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
	Research	5	333445555	1988-05-22
	Administration	4	987654321	1995-01-01
	Headquarters	1	888665555	1981-06-19

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

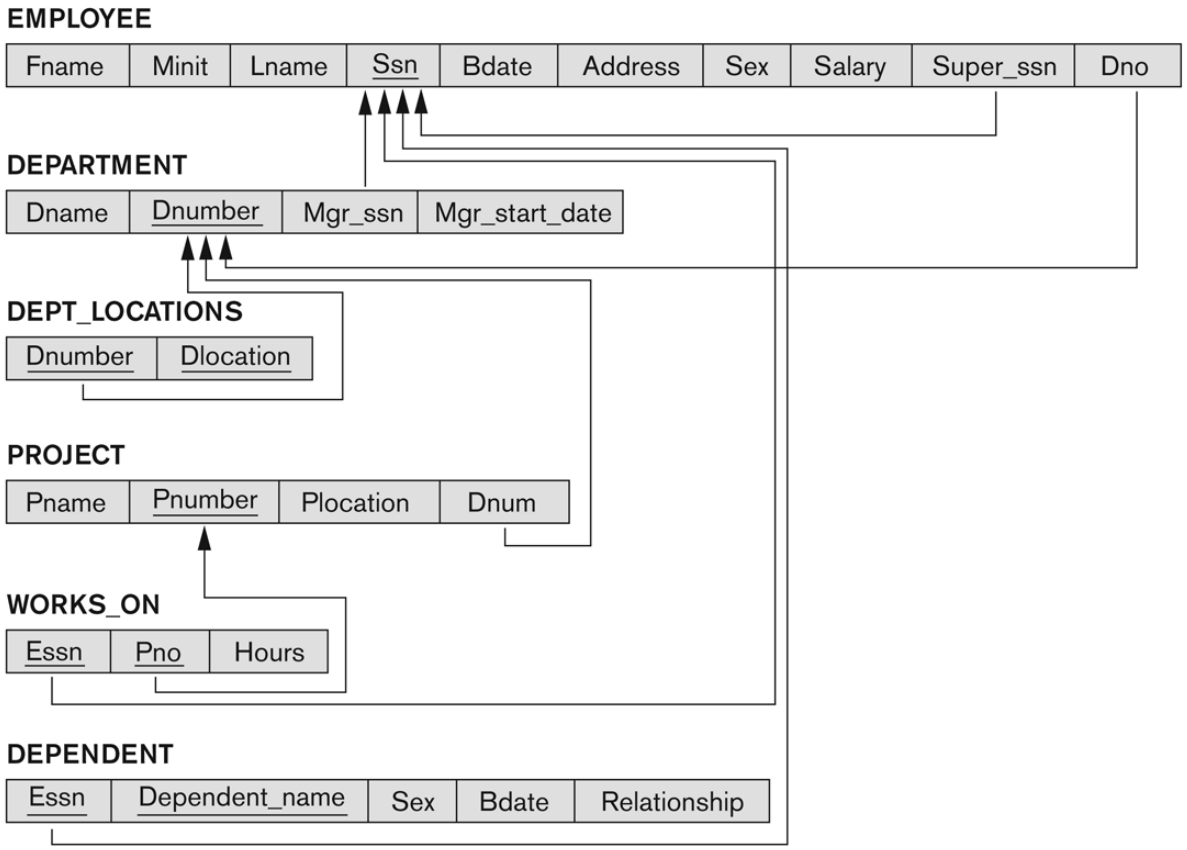
DEPENDENT	ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP
	333445555	Alice	F	1986-04-05	DAUGHTER
	333445555	Theodore	M	1983-10-25	SON
	333445555	Joy	F	1958-05-03	SPOUSE
	987654321	Abner	M	1942-02-28	SPOUSE
	123456789	Michael	M	1988-01-04	SON
	123456789	Alice	F	1988-12-30	DAUGHTER
	123456789	Elizabeth	F	1967-05-05	SPOUSE

entity integrity and referential integrity

- **entity integrity** constraint: **no primary key value can be null**
- **referential integrity** constraint
  - specified **between two relations** and is used to maintain the **consistency among tuples** in the two relations
  - a tuple in one relation that refers to another relation must refer to an **existing** tuple in that relation
- **foreign key** (FK) between R_1 and R_2
  - a set of attributes FK in relation schema R_1 is a **foreign key** of R_1 that references relation R_2 if it satisfies the following:
    - the attributes in FK have the **same domain(s)** as the **primary key attributes** PK of R_2; the attributes FK are said to reference the relation R_2
    - a 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 **null**. in the former case, we have $t_1[\text{FK}] = t_2[\text{PK}]$, and we say that the tuple t_1 references the tuple t_2
  - if these two conditions hold, a referential integrity constraint from R_1 to R_2 is said to hold

example

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



- a value of **DNO** in any tuple t_1 of the EMPLOYEE relation must match a value of the primary key of DEPARTMENT – the DNUMBER attribute – in some tuple t_2 of the DEPARTMENT relation, or the value of DNO can be null if the employee does not belong to a department
- a foreign key can refer to its own relation -> the attribute SUPERSSN in EMPLOYEE

other types of constraints

- **semantic integrity** constraints
  - specified and enforced **within the application programs** or by a general purpose constraint specification language
  - example: the salary of an employee should not exceed the salary of the employee's supervisor
- **functional dependency** constraint
  - establishes a functional relationship **among two sets of attributes *X* and *Y***
  - specifies that the value of *X* determines the value of *Y* in all states of a relation
- **transition constraints** (as opposed to the state constraints)
  - defined to deal with state changes in the DB
  - e.g., the salary of an employee can only increase

dealing with constraint violations: insert

- insert operation
  - can violate any of the four types of constraints discussed so far: domain, key, entity integrity, referential integrity constraints
- examples
  - insert <‘Cecilia’, ‘F’, ‘Kolon’, null, ‘1960-4-5’, ‘6767 Windy Lane’, F, 28000, null, 4> -> violates the entity integrity constraint!
  - insert <‘Alicia’, ‘J’, ‘Zelaya’, ‘999887777’, ‘1960-4-5’, ‘6767 Windy Lane’, F, 28000, ‘987654321’, 4> -> violates the key constraint!
  - insert <‘Cecilia’, ‘F’, ‘Kolon’, ‘677678989’, ‘1960-4-5’, ‘6767 Windy Lane’, F, 28000, ‘987654321’, 100>

    -> violates the referential integrity constraint!

DNUMBER	DLOCATION
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	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	1968-07-19	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

dealing with constraint violations: delete

- delete operation
  - can violate **only referential integrity**, if the tuple being deleted is referenced by the foreign keys from other tuples in the DB
- examples
  - delete the EMPLOYEE tuple with SSN = '999887777'
  - delete the EMPLOYEE tuple with SSN = '333445555'

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

DEPARTMENT	DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
	Research	5	333445555	1988-05-22
	Administration	4	987654321	1995-01-01
	Headquarters	1	888665555	1981-06-19

EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	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	1968-07-19	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

dealing with constraint violations: update

- update operation
  - updating an attribute that is neither a primary key nor a foreign key usually causes no problems
  - **modifying** a primary key value is similar to **deleting** one tuple and **inserting** another in its place
  - if a foreign key attribute is modified, the DBMS must make sure that the new value refers to an existing tuple in the referenced relation (or is null)
- examples
  - update the DNO of the EMPLOYEE tuple with SSN = '999887777' to 7 -> violates the referential integrity constraint!
  - update the SSN of the EMPLOYEE tuple with SSN = '999887777' to '987654321' -> violates the key & referential integrity constraints!

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

DEPARTMENT	DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
	Research	5	333445555	1988-05-22
	Administration	4	987654321	1995-01-01
	Headquarters	1	888665555	1981-06-19

EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	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	1968-07-19	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