Integrity Constraints in the Relational Model

Definition. Integrity constraints are logical statements that restrict the set of allowable relations in a database.

Example.

- database schema, $\mathbf{R} = \{\text{EMP}, \text{DEPT}\}$, with
- schema(EMP) = {ENAME, DNAME, ADDRESS, POSTCODE, LOC}
- schema(DEPT) = {DNAME, MNAME, NO_EMPS, LOC}.
- database, d = $\{r_1, r_2\}$ OVER $\mathbf{R}$,
- r_1 is a relation over EMP, and
- r_2 is a relation over DEPT.

Functional Dependencies

- Stating that ENAME is a **key** of EMP, means that no two distinct tuples in r_1 have the same ENAME.
- Stating that DNAME is a **key** of DEPT, means that no two distinct tuples in r_2 have the same DNAME.

  ⋆ Keys are special cases of **Functional Dependencies** (**FDs**).
- An example of an FD which is not the result of a key, is the constraint that an ADDRESS has a unique POSTCODE.

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Go Back

Full Screen

Close

Quit

Inclusion Dependencies

- Stating that DNAME in EMP is a foreign key referencing the key DNAME in DEPT, means that whenever there is a tuple in r_1 with a nonnull DNAME-value, say *mark*, then there is a corresponding tuple in r_2 whose DNAME-value is also *mark*.

  ⋆ Foreign keys are special cases of **Inclusion Dependencies** (**INDs**).

- An example of an IND which is not the result of a foreign key, is the constraint that the LOCation an employee works in is included in the LOCations of the departments.

Definition. Constraints that depend on the equality or inequality of values in tuples of relations are called **data dependencies**.

- FDs and INDs are data dependencies.

Definition. Constraints that restrict the allowable domain values are called **domain dependencies** (**DDs**).

- An example of a DD is that SALARY ranges between 15 and 40.
- Another example of a DD is that ENAME is a string of at most 25 characters.

Definition. Constraints that restrict the cardinality of a projection of a relation onto a set of attributes are called **cardinality constraints** (**CCs**).

- An example of a CC is that there should not be more managers than employees.
- Another example of a CC that the number of students doing the MSc course should not exceed 100.

An example database.

R is a relation schema, with
schema(R) = {ENAME, DNAME, MNAME}
Relation r over R is given by

ENAME	*DNAME*	*MNAME*
Mark	*Computing*	*Steve*
Angela	*Computing*	*Steve*
Graham	*Computing*	*Steve*
Paul	*Math*	*Donald*
George	*Math*	*Donald*

S is a relation schema, with
schema(S) = {ENAME, CNAME, SAL}

Relation s over S is given by

$ENAME$	$CNAME$	SAL
$Jack$	$Jill$	25
$Jack$	$Jake$	25
$Jack$	$John$	25
$Donald$	Dan	30
$Donald$	$David$	30

Functional Dependencies

Definition. A FD over R is a statement of the form

$$\mathrm{R} : \mathrm{X} \rightarrow \mathrm{Y} \text{ (or simply } \mathrm{X} \rightarrow \mathrm{Y})$$

where X and Y are subsets of schema(R).

- $\mathrm{R} : \{\mathrm{ENAME}\} \rightarrow \{\mathrm{DNAME,MNAME}\}$,
  an employee works in one department and has one manager.
- $\mathrm{S} : \{\mathrm{ENAME}\} \rightarrow \{\mathrm{SAL}\}$,
  an employee has one salary.

Definition. A FD X → Y is **satisfied** in a relation r, if whenever two rows in r have the same X-value they also have the same Y-value.

Alternative definition. An FD X → Y is **satisfied** in a relation r, if for each X-value of r there is at most one Y-value.

NAME → AGE is satisfied in the following relation:

NAME	*AGE*	*CHILD*
Jack	20	*John*
Jack	20	*Jane*

NAME → AGE is violated in the following relation:

NAME	*AGE*	*CHILD*
Jack	20	*John*
Jack	30	*Jane*

Definition. A relation r over R **satisfies** a set of FDs F over R, if all the FDs in F are satisfied in r.

Let $F_1 = \{\{\text{ENAME}\} \rightarrow \{\text{DNAME}\}, \{\text{DNAME}\} \rightarrow \{\text{MNAME}\}\}$ be a set of FDs over R_1.
It can be verified that r satisfies F_1.

Let $F_2 = \{\text{ENAME} \rightarrow \text{SAL}\}$ be a set of FDs over R_2.
It can be verified that s satisfies F_2.

An example of one FD satisfied and the other violated.

NAME → AGE is satisfied in the following relation, but AGE → CHILD and NAME → CHILD are violated:

NAME	*AGE*	*CHILD*
Jack	20	*John*
Jack	20	*Jane*

Definition. A set of attributes X contained in schema(R) is a **superkey** for a relation r over R if r satisfies X $\rightarrow$ schema(R).

Definition. A set of attributes X contained in schema(R) is a **key** for r if

(i) X is a superkey for R, and

(ii) for **no** proper subset Y of X, is Y a superkey for r.

⋆ What are the superkeys and keys for r and s ?

Let schema(R) = SPJ :

- S stands for student,
- J stands for subject and
- P stands for position.

Let F be the following set of FDs over R :

- $SJ \rightarrow P$, i.e. every student has one position in each subject.
- $PJ \rightarrow S$, i.e. every position has one student in each subject.

⋆ What are the superkeys and keys of relations over R that satisfy F ?

Home Page
Title Page
◀◀ ▶▶
◀ ▶
Page 14 of 22
Go Back
Full Screen
Close
Quit

A comprehensive example.

Let $\mathcal{R}$ be a relation schema with $\text{schema}(\mathcal{R}) = \text{CTHRSG}$:

- C stands for a course,
- T stands for a teacher,
- H stands for hour,
- R stands for room,
- S stands for student and
- G stands for grade.

$\mathcal{F}$ is the following a set of FDs over $\mathcal{R}$:

- C → T, i.e. a course has one teacher.
- HR → C, i.e. a room can only only have one course at any time.
- HT → R, i.e. a teacher can only be in one room at any time.
- CS → G, i.e. a student has one grade per course.
- HS → R, i.e. a student can only be in one room at any time.

⋆ What are the superkeys and keys for this example ?

Definition. The **closure** of X with respect to F, denoted by CLOSURE(X, F), is the sets of all attributes Y such that Y can be shown to be dependent on X.

The closure of X is computed by the following algorithm:

Algorithm 1 (CLOSURE(X, F))

```
begin
   Cl := X;
   Done := false ;
   while not Done do
      Done := true;
      for each $W \rightarrow Z$ in F do
         if W is a subset of Cl and Z is not a subset of Cl then
            Cl := Cl union Z;
            Done: = false;
         end if
      end for
   end while
   return Cl;
end.
```

For our example we have:

- CLOSURE(C, $\mathcal{F}$) = CT.
- CLOSURE(HR, $\mathcal{F}$) = HRCT.
- CLOSURE(RSG, $\mathcal{F}$) = RSG.
- CLOSURE(HRSG, $\mathcal{F}$) = HRSGCT.

Alternative Definition. A set of attributes X contained in schema(R) is a **superkey** for R with respect to F if

$$\text{CLOSURE(X, F)} = \text{schema(R)}.$$

$\star$ what are the keys and superkeys of our comprehensive example?

Let R be a relation schema with schema(R) $= \{A_1, A_2, A_3, B_1, B_2, B_3, C\}$.

Let F $= \{A_1 \rightarrow B_1, A_2 \rightarrow B_2, A_3 \rightarrow B_3, B_1 \rightarrow A_1, B_2 \rightarrow A_2, B_3 \rightarrow A_3, \{B_1, B_2, B_3\} \rightarrow C\}$,

$\star$ How many keys does R have with respect to F and what are they?

Definition. A set of FDs F over R is a **cover** of a set of FDs G over R if they are equivalent with respect to the closures of the left-hand sides of their FDs.

Informal Definition. A set of FDs F is a **cover** of a set of FDs G if their semantics are equivalent, i.e. either F and G can be used to model the application.

Example.

Let schema(R) $=$ EPL :

- E stands for employee,
- P stands for project and
- L stands for location.

Let F_1 be the following set of FDs over R :

- E $\rightarrow$ P, ie. an employee works on one project.
- P $\rightarrow$ L, i.e. a project is situated in one location.

$G_1 = \{\mathrm{E} \rightarrow \mathrm{P}, \mathrm{P} \rightarrow \mathrm{L}, \mathrm{E} \rightarrow \mathrm{L}\}$ is a cover of F_1.

$\star$ G_1 has the **redundant** FD E $\rightarrow$ L, which can be derived from F_1.

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Go Back

Full Screen

Close

Quit

Let $F_2 = \{\mathrm{E} \longrightarrow \mathrm{PL}\}$ be a set of FDs over R.

$\star$ F_2 is *not* a cover of F_1, why?

$G_3 = \{\mathrm{E} \longrightarrow \mathrm{P},\ \mathrm{E} \longrightarrow \mathrm{L}\}$ is a cover of F_2.

$\star$ G_3 has *more* FDs than F_2, i.e. it is *not* **minimum**.

$G_4 = \{\mathrm{E} \longrightarrow \mathrm{P},\ \mathrm{EP} \longrightarrow \mathrm{L}\}$ is a cover of F_2.

$\star$ The FD EP $\longrightarrow$ L is *not* **reduced**, since E $\longrightarrow$ L can be derived from G_4.

Let schema(R) = TNOML :

- T stands for teacher social security number,
- N stands for teacher name,
- O stands for office number,
- M stands for day and hour ,
- L stands for lecture theatre code.

Let F be the following set of FDs over R :

- $NO \rightarrow T$, a name and office is associated with one teacher.
- $T \rightarrow NO$, i.e. a teacher has one name and office.
- $TM \rightarrow L$, i.e. a teacher is in one lecture at a given time.

$G = \{NO \rightarrow T, T \rightarrow NO, NOM \rightarrow L\}$ is a cover of F.

⋆ The number of attributes in G is *greater* than the number of attributes in F, i.e. it is *not* **optimum**.