Integrity Constraints in the Relational Model

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

Example.

- database schema, $\mathbf{R} = \{EMP, DEPT\}$, 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 $schema(EMP)$, and
- r_2 is a relation over $schema(DEPT)$.

Keys and Functional Dependencies (FDs)

- 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*.

Inclusion Dependencies (INDs)

- 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 *Sales*, then there is a corresponding tuple in r_2 whose *DNAME*-value is also *Sales*.
- 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 *LOC*ation an employee works in is included in the *LOC*ations of the departments.

An example database

Relation schemas *R* and *S*, and relations *r* and *s*

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

Database Management
Peter Wood
Integrity Constraints
Functional Dependencies
Keys and Superkeys
Closure Algorithm

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

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

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

We say that X **functionally determines** Y.

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

Database Management
Peter Wood
Integrity Constraints
Functional Dependencies
Keys and Superkeys
Closure Algorithm

Definition. An FD $X \to Y$ 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 $X \to Y$ is **satisfied** in a relation r, if for each X-value of r there is at most one Y-value.

$NAME \to AGE$ is satisfied in the following relation:

NAME	AGE	CHILD
Jack	20	John
Jack	20	Jane

$NAME \to AGE$ is **violated** in the following relation:

NAME	AGE	CHILD
Jack	20	John
Jack	30	Jane

Database Management
Peter Wood
Integrity Constraints
Functional Dependencies
Keys and Superkeys
Closure Algorithm

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 $F_1 = \{\{ENAME\} \to \{DNAME\}, \{DNAME\} \to \{MNAME\}\}$ be a set of FDs over R_1.

It can be verified that r satisfies F_1.

Let $F_2 = \{ENAME \to SAL\}$ 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.

$NAME \rightarrow AGE$ is satisfied in the following relation, but $AGE \rightarrow CHILD$ and $NAME \rightarrow CHILD$ are violated:

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

Database Management
Peter Wood
Integrity Constraints
Functional Dependencies
Keys and Superkeys
Closure Algorithm

Definition. An FD $X \rightarrow Y$ **holds** on relation schema R if every allowable (legal) relation r over R satisfies $X \rightarrow Y$.

Definition. A set F of FDs **holds** on relation schema R if every allowable (legal) relation r over R satisfies each FD in F.

Here "allowable" and "legal" mean that the relations would correctly model some part of the real world.

Database Management
Peter Wood
Integrity Constraints
Functional Dependencies
Keys and Superkeys
Closure Algorithm

Problem for you to work on

Consider the following relation about films:

Title	Year	Genre	StarName
Star Wars	1977	SciFi	Carrie Fisher
Star Wars	1977	SciFi	Harrison Ford
Raiders ...	1981	Action	Harrison Ford
Raiders ...	1981	Adventure	Harrison Ford
When Harry ...	1989	Comedy	Carrie Fisher

1. What FDs are *satisfied* by the above relation?
2. What FDs are *violated* by the above relation?
3. What FDs would you expect to *hold* on the above relation schema?

Database Management
Peter Wood
Integrity Constraints
Functional Dependencies
Keys and Superkeys
Closure Algorithm

Keys and Superkeys

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

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

1. X is a superkey for R, and
2. for **no** proper subset Y of X, is Y a superkey for R.

$\star$ What are the superkeys and keys for our earlier R and S about employees ?

Let $schema(R) = SPJ$ (we often abbreviate the set notation $\{S, P, J\}$):

- 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 keys of relations over $schema(R)$ that satisfy F ?

Database Management
Peter Wood
Integrity Constraints
Functional Dependencies
Keys and Superkeys
Closure Algorithm

A comprehensive example.

Let U be a relation schema with $schema(U) = CTHRSG$:

- 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.

Database Management
Peter Wood
Integrity Constraints
Functional Dependencies
Keys and Superkeys
Closure Algorithm

F is the following a set of FDs over U :

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

What are the keys for this example ?

Database Management
Peter Wood
Integrity Constraints
Functional Dependencies
Keys and Superkeys
Closure Algorithm

Closure of a set of attributes

The **closure** of set of attributes X with respect to F, denoted by $CLOSURE(X, F)$, is the set Y of all attributes such that X functionally determines Y.

The closure of X is computed by the following algorithm:

```
CI := X;
Done := false ;
while not Done do
  Done := true;
  for each W → Z in F do
    if W is a subset of CI and Z is not a subset of CI then
      CI := CI union Z;
      Done: = false;
    end if
  end for
end while
return CI;
```

For our example we have:

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

Superkeys

Database Management
Peter Wood
Integrity Constraints
Functional Dependencies
Keys and Superkeys
Closure Algorithm

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

$$CLOSURE(X, F) = schema(R).$$

⋆ what are the keys of our comprehensive example?

Database Management
Peter Wood
Integrity Constraints
Functional Dependencies
Keys and Superkeys
Closure Algorithm

Problem for you to work on

Consider relation schema R where $schema(R) = ABCD$.

Let the set F of FDs which hold on R be $\{AB \rightarrow C, C \rightarrow D, D \rightarrow A\}$.

1. Compute
   1.1 $CLOSURE(A, F)$
   1.2 $CLOSURE(B, F)$
   1.3 $CLOSURE(C, F)$
   1.4 $CLOSURE(D, F)$.
2. What are all the keys of R?
3. What are all the superkeys for R which are not keys?

Database Management
Peter Wood
Integrity Constraints
Functional Dependencies
Keys and Superkeys
Closure Algorithm

Problem for you to work on

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

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

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