Integrity Constraints in the Relational Model

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

**Example.**

- database schema, \( R = \{\text{EMP}, \text{DEPT}\} \), with
  - \( \text{schema}(\text{EMP}) = \{\text{ENAME}, \text{DNAME}, \text{ADDRESS}, \text{POSTCODE}, \text{LOC}\} \)
  - \( \text{schema}(\text{DEPT}) = \{\text{DNAME}, \text{MNAME}, \text{NO\_EMPS}, \text{LOC}\} \)
- database, \( d = \{r_1, r_2\} \) over \( R \),
- \( r_1 \) is a relation over \( \text{EMP} \), and
- \( r_2 \) is a relation over \( \text{DEPT} \).

Inclusion Dependencies (INDs)

- Stating that \( \text{DNAME} \) in \( \text{EMP} \) is a foreign key referencing the key \( \text{DNAME} \) in \( \text{DEPT} \), means that whenever there is a tuple in \( r_1 \) with a nonnull \( \text{DNAME} \)-value, say \( \text{Sales} \), then there is a corresponding tuple in \( r_2 \) whose \( \text{DNAME} \)-value is also \( \text{Sales} \).
- 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 \( \text{LOC} \)ation an employee works in is included in the \( \text{LOC} \)ations of the departments.

Keys and Functional Dependencies (FDs)

- Stating that \( \text{ENAME} \) is a key of \( \text{EMP} \), means that no two distinct tuples in \( r_1 \) have the same \( \text{ENAME} \).
- Stating that \( \text{DNAME} \) is a key of \( \text{DEPT} \), means that no two distinct tuples in \( r_2 \) have the same \( \text{DNAME} \).
- 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 \( \text{ADDRESS} \) has a unique \( \text{POSTCODE} \).

Data and Domain Dependencies

**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 \( \text{SALARY} \) ranges between 15 and 40.
- Another example of a DD is that \( \text{ENAME} \) is a string of at most 25 characters.
Cardinality Constraints

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 is that the number of students doing the MSc course should not exceed 100.

$S$ is a relation schema, with $\text{schema}(S) = \{\text{ENAME, CNAME, SAL}\}$
Relation $s$ over $S$ is given by

<table>
<thead>
<tr>
<th>ENAME</th>
<th>CNAME</th>
<th>SAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jack</td>
<td>Jill</td>
<td>25</td>
</tr>
<tr>
<td>Jack</td>
<td>Jake</td>
<td>25</td>
</tr>
<tr>
<td>Jack</td>
<td>John</td>
<td>25</td>
</tr>
<tr>
<td>Donald</td>
<td>Dan</td>
<td>30</td>
</tr>
<tr>
<td>Donald</td>
<td>David</td>
<td>30</td>
</tr>
</tbody>
</table>

An example database

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

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

<table>
<thead>
<tr>
<th>ENAME</th>
<th>DNAME</th>
<th>MNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mark</td>
<td>Computing</td>
<td>Steve</td>
</tr>
<tr>
<td>Angela</td>
<td>Computing</td>
<td>Steve</td>
</tr>
<tr>
<td>Graham</td>
<td>Computing</td>
<td>Steve</td>
</tr>
<tr>
<td>Paul</td>
<td>Math</td>
<td>Donald</td>
</tr>
<tr>
<td>George</td>
<td>Math</td>
<td>Donald</td>
</tr>
</tbody>
</table>

Functional Dependencies

Definition. An FD over $R$ is a statement of the form $R : X \rightarrow Y$ (or simply $X \rightarrow Y$)
where $X$ and $Y$ are subsets of $\text{schema}(R)$.
We say that $X$ functionally determines $Y$.

▶ $R : \{\text{ENAME}\} \rightarrow \{\text{DNAME, MNAME}\}$,
each employee works in one department and has one manager.
▶ $S : \{\text{ENAME}\} \rightarrow \{\text{SAL}\}$,
each employee has one salary.
**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:

<table>
<thead>
<tr>
<th>NAME</th>
<th>AGE</th>
<th>CHILD</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>20</td>
<td>Jane</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>NAME</th>
<th>AGE</th>
<th>CHILD</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>20</td>
<td>Jane</td>
</tr>
</tbody>
</table>

An example of one FD satisfied and the other violated. $NAME \to AGE$ is satisfied in the following relation, but $AGE \to CHILD$ and $NAME \to CHILD$ are violated:

<table>
<thead>
<tr>
<th>NAME</th>
<th>AGE</th>
<th>CHILD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jack</td>
<td>20</td>
<td>John</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>NAME</th>
<th>AGE</th>
<th>CHILD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jack</td>
<td>20</td>
<td>John</td>
</tr>
</tbody>
</table>

**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 $F_2$.

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

**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.
Problem for you to work on

Consider the following relation about films:

<table>
<thead>
<tr>
<th>Title</th>
<th>Year</th>
<th>Genre</th>
<th>StarName</th>
</tr>
</thead>
<tbody>
<tr>
<td>Star Wars</td>
<td>1977</td>
<td>SciFi</td>
<td>Carrie Fisher</td>
</tr>
<tr>
<td>Star Wars</td>
<td>1977</td>
<td>SciFi</td>
<td>Harrison Ford</td>
</tr>
<tr>
<td>Raiders ...</td>
<td>1981</td>
<td>Action</td>
<td>Harrison Ford</td>
</tr>
<tr>
<td>Raiders ...</td>
<td>1981</td>
<td>Adventure</td>
<td>Harrison Ford</td>
</tr>
<tr>
<td>When Harry ...</td>
<td>1989</td>
<td>Comedy</td>
<td>Carrie Fisher</td>
</tr>
</tbody>
</table>

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?

Let \( \text{schema}(R) = SPJ \) (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 \( R \):
- \( SJ \to P \), i.e. every student has one position in each subject.
- \( PJ \to S \), i.e. every position has one student in each subject.

What are the superkeys and keys of relations over \( R \) that satisfy \( F \) ?

Keys and Superkeys

**Definition.** A set of attributes \( X \) contained in \( \text{schema}(R) \) is a superkey for a relation schema \( R \) if \( X \to \text{schema}(R) \) holds on \( R \).

**Definition.** A set of attributes \( X \) contained in \( \text{schema}(R) \) 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 \( R \).

* What are the superkeys and keys for \( R \) and \( S \) ?

A comprehensive example.

Let \( U \) be a relation schema with \( \text{schema}(U) = CTHRSG \):
- \( C \) 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.
$F$ 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 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 superkeys and keys for this example?

For our example we have:

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

---

**Closure of a set of attributes**

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

The closure of $X$ is computed by the following algorithm:

1. $Cl := X$;
2. $Done := false$;
3. **while not Done do**
4. $Done := true$;
5. **for each** $W \rightarrow Z$ in $F$ **do**
6.  **if** $W$ is a subset of $Cl$ and $Z$ is not a subset of $Cl$ **then**
7.  $Cl := Cl$ union $Z$;
8.  $Done := false$;
9.  **end if**
10. **end for**
11. **end while**
12. **return** $Cl$;

---

**Superkeys**

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

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

* what are the keys and superkeys of our comprehensive example?
Problem for you to work on

Consider relation schema $R$ where $\text{schema}(R) = ABCD$.
Let the set $F$ of FDs which hold on $R$ be
\{ $AB \rightarrow C, C \rightarrow D, D \rightarrow A$ \}.

1. Compute
   1.1 $\text{CLOSURE}(A, F)$
   1.2 $\text{CLOSURE}(B, F)$
   1.3 $\text{CLOSURE}(C, F)$
   1.4 $\text{CLOSURE}(D, F)$.

2. What are all the keys of $R$?
3. What are all the superkeys for $R$ which are not keys?

Problem for you to work on

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