

7.2 Static Integrity Constraints

Type integrity / domain constraints

- ❑ Each attribute value must be atomic.
- ❑ determination of upper and lower bounds of attribute values
 - SQL92 allows the explicit definition of domains (not implemented in Oracle)
- ❑ enumeration of values
 - Such enumeration types can be modeled by an own relation with one attribute.
- ❑ null values
 - The construct **not null** attached to an attribute declaration can be used to achieve that the value of the corresponding attribute exists for all tuples.

Tuple constraints

- ❑ restriction of the values which a tuple can take with respect to its different attributes
- ❑ example: In a relation *occupancy*, where a tuple stores the reservation of a seat from a train station X to a station Y , X should be unequal to Y .
- ❑ In SQL a tuple constraint can be defined by the **check** clause during the specification of the schema.

Relation constraints / key constraints

- ❑ refers to the set of all tuples of a relation
- ❑ **key integrity** by means of primary key and candidate key
 - no two tuples with equal keys
 - in SQL declaration of this constraint with **unique** and **primary key** during the specification of the schema
 - No primary key or candidate key can have the value **null**.
- ❑ aggregation constraints
 - example: An aggregate value shall be above or below a predefined bound.
- ❑ Recursive constraints
 - example: In a relation *TrainConnection* each station should be connected with each other station.

Referential constraints

- ❑ refer to several relations

7.3 Referential Integrity

Features

- ❑ **Referential ICs** are conditions for relations which especially model a relationship.
- ❑ They are used to maintain the consistency between tuples of two relations.
- ❑ A referential IC requires that a tuple of a relation that refers to another relation has to refer to an *existing* tuple of that relation.
- ❑ Let R and S be relations with the schemas R and S . Let $K \subseteq R$ be a primary key of R . Then $F \subseteq S$ is called **foreign key** of S , if for each data record s of relation S one of the following conditions holds:
 - $\forall A \in F : s[A] = \text{null}$.
 - There is a data record r of R such that $s[F] = r[K]$.
- ❑ example: relations *customers*, *products*, *orders*
(*m:n*-relationship between *customers* and *products*)
- ❑ possible problems if referential integrity is not fulfilled:
 - Customer orders products which do not exist
 - Products can be ordered by a customer who does not exist.

Maintenance of referential integrity

- ❑ In a relation, which models a relationship, we should ensure that values are assigned to the foreign keys.
- ❑ formulation of explicit integrity constraints in the query language, small extensions of the query language
- ❑ formulation in relational algebra:
 - relation R with primary key K
 - relation S with foreign key F that references K
$$\pi_F(S) \subseteq \pi_K(R)$$
- ❑ Allowed changes
 - insertion of a tuple s into S if $s[F] \in \pi_K(R)$
 - change of an attribute value of a tuple s if $s'[F] \in \pi_K(R)$ (let s' be the new value of tuple s)
 - change of $r[K]$ of a tuple r if $\sigma_{F=r[K]}(S) = \emptyset$
 - deletion of a tuple r of R if $\sigma_{F=r[K]}(S) = \emptyset$

Referential integrity in the ER model

- ❑ N -ary relationships contain the keys of all participating entities, i.e., all keys are foreign keys.
- ❑ A weak entity set must include the primary key of the entity set on which the weak entity set depends. That is, the weak entity set contains the primary key of the strong entity set as foreign key.

Integrity constraints in SQL

- ❑ domain constraints
 - **not null**
 - + null value as attribute value not permitted
 - + example: `FamilyName character(20) not null`
 - **default** <value>
 - + initialization of an attribute value with the value <value> which must be a constant or the result of the evaluation of an expression. The value <value> must be compatible with the domain of the attribute.
 - + example: `NoOfChildren smallint default 0`
 - **default null**: optional, default value

Tuple constraint

- ❑ **check** clause as an integrity constraint that can be represented as a simple Boolean expression
- ❑ typical applications: range queries, realization of enumeration types, handling of null values
- ❑ example for range query: limitation that students may study for a maximum of 8 semesters.
check sem **between** 1 **and** 8
- ❑ example for enumeration type:
check rank **in** ("C2", "C3", "C4")
- ❑ handling of null values
 - Null value fits for **check** or foreign key constraints to arbitrary values. If a part of a foreign key is **null**, the key relates to all data which have the same defined part.
 - If a foreign key is to be totally **null** oder totally defined, this must be explicitly expressed by a **check** clause.

- example

check ((S_1 is null and S_2 is null and ...) or
(S_1 is not null and S_2 is not null and ...))

- The constraint

check sem **between** 1 **and** 8

is also fulfilled for a null value as semester.

Relation constraints/ key constraints

□ **primary key** clause

- automatically specified as **not null**
- example:

FirstName *character*(20) **not null**,

LastName *character*(20) **not null**,

Salary *numeric*(11,2),

primary key (LastName, FirstName)