Contents

Integrity vs Security

- Integrity and Security are related but they are not the same thing
  - Integrity is concerned with *accidental* corruption
  - Security is concerned with *deliberate* corruption
- Integrity
  - Integrity Constraints
- Security
  - Security Policies
  - Access Control

Relational Model Constraints

- Constraints expressed in the Relational Schema
  - *Explicit Constraints*
- Constraints that cannot be expressed in the Relational Schema
  - *Semantic Constraints*
  - Can be expressed in SQL in some cases
  - Usually enforced by the application programs

Itegrity Constraints

- Three types of integrity constraints are considered part of the relational model:
  - Key
  - Entity Integrity
  - Referential Integrity
- The DBMS must be able to enforce these constraints

Key Constraint

- Specifies that there may not be any ducplicate entries in key attributes
  - Primary Key
  - Candidate Keys
- Keys are used to uniquely identify a tuple
  - Having a duplicate valye in a Key implies that we cannot uniquely identify some tuples

Entity Integrity Constraint

- Specifies that there may not be any NULL values in the Primary Key attribute
  - The Primary Key is used to uniquely identify each tuple in a relation
  - Having a NULL in a Primary Key implies that be cannot identify some tuples

Referential Integrity

- Entity constraints are specified on individual relations
- Referential Integrity constraints are specified between two relations
  - Maintains consistency among tuples in the two relations
  - A tuple in one relation that refers to another relation, must refer to an *existing typle* in that relation
- A Foreign Key formally specifies a Referential Integrity Constraint between two relations

NULL Keys

- As per the Entity Integrity constraint
  - No part of a Primary Key can be NULL
- However, Foreign Keys in certain circumstances may be NULL
  - A decision must be made during schema design as to where it is valid for the foreign key to be NULL at any point

Referential Integrity

- When defining an attribute as a Foreign Key
  - You must also specify whether or not the foreign key is allowed to contain NULLs
- In the case of a composite Foreign Key
  - If the Foreign Key is allowed to contain NULLs
  - Then either all the component attributes should be NULL or none of them NULL
    - In order to enforce referential integrity

Constraint Violation

- There are three basic operations that modify that state of relations in a DB
  - Insert
  - Update
  - Delete
- These operations should not violate the integrity constraints specified for the DB
  - Key, Entity, Referential

Insert

- Insert provides a list of attribute values for a new tuple t that is to be added to relation R
- Insert can violate all the integrity constraints that we have discussed
  - Key
  - Entity Integrity
  - Referential Integrity

Delete

- To specify a deletion, a condition on the attributes of a relation is created which selections one or more tuples to be deleted
- The Delete operation can only violate the Referential Integrity constraint

Cascading Deletes

- An option to address Delete operations which violate Referential Integirty is to *cascade*, or propagate, the deletion
- The DBMS could automatically delete the offending tuples from WORKS_ON
  - In addition to the original tuple in EMPLOYEE
  - This must be implemented carefully, as it can lead to unintential loss of data

Update

- An update operation is used to change the values of one or more attributes of a relation
- To specify an update, a condition on the attributes of a relation is created which selects one or more tuples to be modified
- Updates can violate all the integrity constraints that we have discused
  - Key
  - Entity Integrity
  - Referential Integrity

Cascading Updates

- As with Delete, an option to address Update operations which violate Referential Integirty is to *cascada*, or propagate, the update
- The DBMS could automatically update the relations which have a Foreign Key to Ssn
  - WORKS_ON, DEPARTMENT, DEPENDENT, EMPLOYEE

Alternatives to Cascading

- The alternatives to the cascading of updates or deletes are
  - Rejection of the update or delete as long as foreign key references exist
  - Update of the corresponding foreign key to NULL
  - Update of the corresponding foreign key to some default value

Constraints in SQL

- Constraints specified as part of relation, or table, definition are called "table constraints"
  - They are specified on each table individually
- They are typically specified during able creation in the CREATE TABLE statement
  - Can be added later using ALTER TABLE
- Constraints that affect more than one table are called Assertions

Primary Key

- The PRIMARY KEY constraints specifies the attribute(s) that form the Primary Key
  - For a single attribute, the constraint can directly follow the attribute specification
    - Dnumber INT PRIMARY KEY
  - Composite keys can be specified at the end of the CREATE TABLE statement
    - PRIMARY KEY (Dnumber, Dlocation)

Unique

- As we have seen, there is often more than one candidiate key in a relation
- Secondary keys can be specified using the UNQIEU constraints
  - For a single attribute, the constraint can directly follow the attribute specification
    - Engine_num INT UNIQUE
  - Composite secondary keys can be specified at the end of the CREATE TABLE statement
    - UNIQUE (Licence_Yr, Licence_Mth, Licence_Day)

Not Null

- By default SQL allows NULLs as attribute values
  - A NOT NULL constraint may be specified if NULLs are not permitted for a specific attribute
  - This is always the case for any attribute that forms part of the Primary Key

```
CREATE TABLE Person
(PPS char(8) NOT NULL PRIMARY KEY,
Fname varchar(255) NOT NULL,
Lname varchar(255),
Phone int);
```

Check

- More complex constraints can be specified using the CHECK clause
  - Used to restrict the values that can be entered for an attribute
- Each CHECK is specified on one or more attributes from a single table
- The CHECK is performed for every tuple that is inserted or modified

Check Clause

- CHECK clauses are specified within the CREATE TABLE statement
- They can be specified on an individual attribute
  - Dnumber INT NOT NULL CHECK (Dnumber > 0 AND Dnumber < 21)
- Or on multiple attributes from the same table
  - CHECK (Dept_create_date <= Mgr_start_date)

Referential Integrity

- Referential Integrity is specified using the FOREIGN KEY clause
  - Specified as the end of the CREATE TABLE statement
    - FOREIGN KEY (Dno) REFERENCES REPARTMENT (Dnumber)
  - Can also have composite Foreign Keys
    - FOREIGN KEY (artist, album) REFERENCES ALBUM (artist, name)

Violation

- As discussed earlier, Referential Integrity can be violated on update, insert or delete
  - Default action in SQL is to reject the operation
- It is possible to specify an alternate action by attaching a clause to each Foreign Key
  - SET NULL
  - CASCADE
  - SET DEFAULT

Naming Constraints

- Constraints can be named using CONSTRAINT
  - Names must be unique within the schema

Assertions

- An Assertion is a stand-alone constraint in a schema
  - Used to specify a restricition that affects more than one table
- Table constraints (using CHECK) can be used to specify multiple table constraints, however, it is better practice to use Assertions:
  - Table constraints are only evaluated if and only if the table to which is it attached has some data
- Assertions are
  - Associated with the relations in question
  - Evaluated before an operation can be performed on those relations
  - Violated if false and the operation is not allowed
  - Define valid states of a DB
  - Actually stored as rows in the ASSERTIONS table which is part of the system catalog

Evaluation

- Assertions are checked at the end of each SQL statement
  - A transaction can be more than one SQL statement
  - Assertion evaluation can be deferred until the end of a transaction, but is always evaluated prior to the completion of a transaction
- If an assertion fails, the DBMS returns an error message and the SQL statement is rejected

Triggers

- Triggers are Event-Condition-Action rules
  - Allow constraints to be checked on specified events and resulting actions to be invoked
- Triggers are only tested when certain events occur
  - e.g. Insert, update, etc.
- When triggered, a specified condition is tested
  - If the condition does not hold, then no further action is taken in response to the event
  - If the condition is satisfied, defined actions associated with the trigger are performed by the DBMS

Assertions vs Triggers

- Assertions
  - Do not modify the data, only check certain conditions
- Triggers
  - Are more powerful because they can check conditions and also modify the data
  - Are linked to specific tales and specific events
- All assertions can be implemented as triggers
- Not all triggers can be implemented as assertions
- Oracle does not have assertions