



CS 220

Database Systems

Fall 2019



Lecture 3

Entity Integrity Constraint



The entity integrity constraint : Primary key value can not be NULL.

Since primary key value is used to identify individual tuples in a relation, having NULL values for the primary key implies that we cannot identify some tuples.

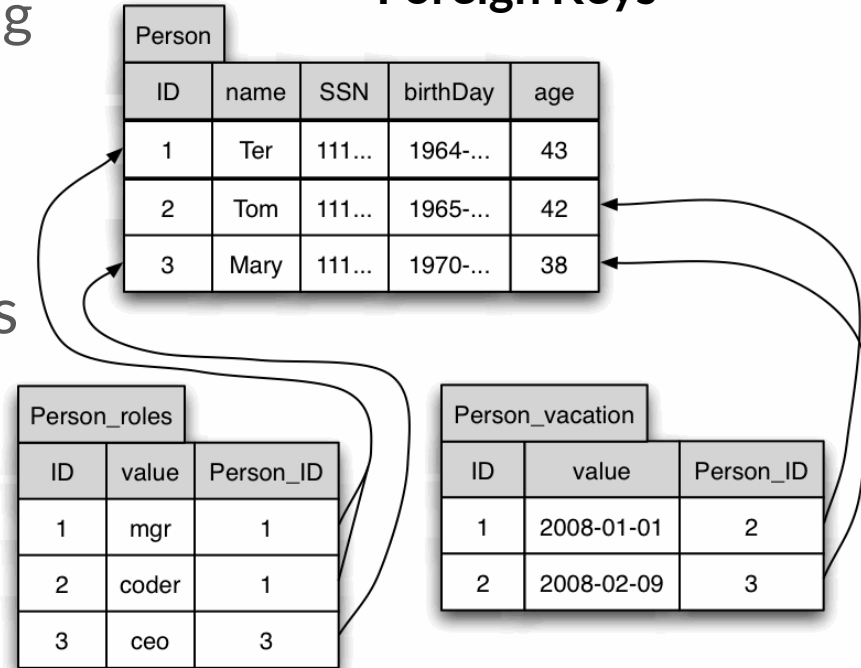
***Key constraints and entity integrity constraints are specified on individual relations.**

Referential Integrity Constraint

- Specified between two relations
- Used to maintain consistency among tuples in the two relations

*Referential integrity constraints typically arise from the relationships among the entities

Foreign Keys



Relational Constraints



Domain Constraint	Atomic and single value from the domain $\text{dom}(A)$
Key Constraint	A key that allows to uniquely identify its tuples
Entity Integrity Constraint	Primary key should always satisfy a NOT NULL constraint
Referential Integrity Constraint	<ul style="list-style-type: none">- A foreign key FK has the same domain as the primary key PK attribute type(s) it refers to.- Either occurs as a value of PK or NULL

General Constraints



There are more general constraints that some DBMSs can enforce.

These constraints are often called enterprise constraints or semantic integrity constraints.

- An employee can not work on more than 2 projects
- An employee must be assigned to at least one project



Structured Query Language

SQL



SQL is the standard query language for relational databases

All major database vendors conform to the SQL standard with minor variations in syntax (different dialects).

SQL is a declarative language (non-procedural). A SQL query specifies what to retrieve but not how to retrieve it.

SQL



Some basic rules for SQL statements:

- There is a set of reserved words that cannot be used as names for database objects. (e.g. SELECT, FROM, WHERE)
- SQL is case-insensitive.
- Only exception is string constants. 'FRED' not the same as 'fred'.
- SQL is free-format and white-space is ignored.
- The semi-colon is often used as a statement terminator, although that is not always required.

SQL



Some basic rules for SQL statements:

- Date and time constants have defined format:
Dates: 'YYYY-MM-DD' e.g. '1975-05-17'
Times: 'hh:mm:ss[.f]' e.g. '15:00:00'
Timestamp: 'YYYY-MM-DD hh:mm:ss[.f]' e.g. '1975-05-17 15:00:00'
- Two single quotes " ' " are used to represent a single quote character in a character constant. e.g. 'Master"s'.

SQL Identifiers



Identifier is the name of the database object

An SQL identifier (name) must follow these rules:

- Only contain upper or lower case characters, digits, and underscore ("_") character
- Be no longer than 128 characters
- DB vendors may impose stricter limits than this.
- Must start with a letter (or underscore) cannot contain spaces
- Note: Quoted or delimited identifiers enclosed in double quotes allow support for spaces and other characters. E.g. "select

SQL Data Types

- In the relational model, each attribute has an associated domain of values.
- In SQL, each column (attribute) has a data type that limits the values that it may store. The standard SQL data types are similar to their programming language equivalents.
- The database will perform (implicit) data type conversion when necessary.
- Explicit data type conversion using functions such as CAST and CONVERT

SQL Data Types



Data Type	Description
BOOLEAN	TRUE or FALSE
CHAR	Fixed length string (padded with blanks) e.g. CHAR(10)
VARCHAR	Variable length string e.g. VARCHAR(50)
BIT	Bit string e.g. BIT(4) can store '0101'
NUMERIC or DECIMAL	Exact numeric data type e.g. NUMERIC(7,2) has a precision (max. digits) of 7 and scale of 2 (# of decimals) e.g. 12345.67
INTEGER	Integer data only
SMALLINT	Smaller space than INTEGER
FLOAT or REAL	Approximate numeric data types. Precision dependent on implementation.
DOUBLE PRECISION	
DATE	Stores YEAR, MONTH, DAY
TIME	Stores HOUR, MINUTE, SECOND
TIMESTAMP	Stores date and time data.
INTERVAL	Time interval.
CHARACTER LARGE OBJECT	Stores a character array (e.g. for a document)
BINARY LARGE OBJECT	Stores a binary array (e.g. for a picture, movie)

SQL User Defined Data Types



The CREATE DOMAIN command allows you to define your own types that are subsets of built-in types:

```
CREATE DOMAIN domainName AS dataType  
    [DEFAULT defaultValue]  
    [CHECK (condition)]
```

Example:

Create user-defined
domain for
Employee.title:

```
CREATE DOMAIN titleType AS CHAR(2)  
DEFAULT 'EE'  
CHECK (VALUE IN  
(NULL,'EE','SA','PR','ME'))
```

SQL User Defined Data Types



The **CHECK** clause can use a nested select statement to retrieve values from the database:

```
CREATE DOMAIN mgrType AS CHAR(5)
    DEFAULT NULL
    CHECK (
        VALUE IN (
            SELECT eno FROM emp
            WHERE title = 'ME' OR title = 'SA'
        )
    );
```

SQL Create Table



The CREATE TABLE command is used to create a table in the database. A table consists of a table name, a set of fields with their names and data types, and specified constraints

```
CREATE TABLE table_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    ....  
);
```


SQL Create Table



The CREATE TABLE command for the Emp relation:

```
CREATE TABLE Emp (  
    eno CHAR(5),  
    ename VARCHAR(30),  
    bdate DATE,  
    title CHAR(2),  
    supereno CHAR(5),  
    dno CHAR(5)  
);
```

SQL Constraints



Constraints are specified in CREATE and ALTER TABLE statements.

Types of constraints

Required data - To specify that a column must always have a data value (cannot be NULL) specify NOT NULL after the column definition.

e.g. eno CHAR(5) NOT NULL

Domain constraints - Used to verify that the value of a column is in a given domain

e.g. title CHAR(2) CHECK (title IN (NULL,'EE','SA','PR','ME'));

Forces the title to be either NULL or one of 4 defined values.

Can also be performed using user-defined types (domains).

SQL Create Table with Constraints



The CREATE TABLE command for the Emp relation:

```
CREATE TABLE Emp (  
    eno CHAR(5),  
    ename VARCHAR(30) NOT NULL,  
    bdate DATE,  
    title CHAR(2),  
    supereno CHAR(5),  
    dno CHAR(5)  
);
```

SQL Constraints- Again

Types of constraints

Tuple constraints - CHECK can also be used on an entire tuple instead of a single attribute

```
CREATE TABLE student(  
    num          CHAR(10)      NOT NULL,  
    honors       CHAR(1),  
    gpa          DECIMAL(3,2),  
    CHECK ( (honors = 'Y' AND gpa > 3.50)  
            OR honors = 'N')  
);
```

SQL Constraints- Again



Types of constraints

Entity Integrity constraints - The primary key of a table must contain a unique, non-null value for each row. The primary key is specified using the PRIMARY KEY clause

There can only be one primary key per relation, other candidate keys can be specified using UNIQUE

SQL Create Table with Entity Integrity Constraints



The CREATE TABLE command for the Emp relation:

```
CREATE TABLE Emp (  
    eno CHAR(5),  
    ename VARCHAR(30) NOT NULL,  
    bdate DATE,  
    title CHAR(2),  
    supereno CHAR(5),  
    dno CHAR(5),  
    PRIMARY KEY (eno)  
);
```

SQL Constraints- Again



Types of constraints

Referential Integrity constraints - Defines a foreign key that references the primary key of another table.

If a foreign key contains a value that is not NULL, that value must be present in some tuple in the relation containing the referenced primary key.

Specify foreign keys using FOREIGN KEY syntax:

FOREIGN KEY (eno) **REFERENCES** Emp(eno)

SQL Referential Integrity & Updates



When you try to INSERT or UPDATE a row in a relation containing a foreign key, that operation is rejected if it violates referential integrity.

When you UPDATE or DELETE a row in the primary key relation, you have the option on what happens to the values in the foreign key relation.

- CASCADE - Delete (update) values in foreign key relation when primary key relation has rows deleted (updated).
- SET NULL - Set foreign key fields to NULL when corresponding primary key relation row is deleted.
- SET DEFAULT - Set foreign key values to their default value (if defined).
- NO ACTION - Reject the request on the parent table

*MySQL equivalent is RESTRICT

SQL Create Table with Referential Integrity Constraints



The CREATE TABLE command for the Emp relation:

```
CREATE TABLE Emp (  
    eno CHAR(5),  
    ename VARCHAR(30) NOT NULL,  
    bdate DATE,  
    title CHAR(2),  
    supereno CHAR(5),  
    dno CHAR(5),  
    PRIMARY KEY (eno),  
    FOREIGN KEY (dno) REFERENCES Dept(dno)  
        ON DELETE SET NULL ON UPDATE CASCADE  
);
```

Example

```
CREATE DOMAIN T_eno AS CHAR(5);
CREATE DOMAIN T_pno AS CHAR(5);
CREATE DOMAIN T_dno AS CHAR(5);

CREATE TABLE Emp(
    eno          T_eno,
    ename        VARCHAR(30) NOT NULL,
    bdate        DATE,
    title        CHAR(2),
    salary       DECIMAL(9,2),
    supereno     T_eno,
    dno          T_dno
    PRIMARY KEY (eno),
    FOREIGN KEY (dno) REFERENCES Dept(dno)
        ON DELETE SET NULL ON UPDATE CASCADE
);
```

Example

```
CREATE TABLE WorksOn (  
    eno    T_eno,  
    pno    T_pno,  
    resp   VARCHAR(20),  
    hours  SMALLINT,  
    PRIMARY KEY (eno,pno),  
    FOREIGN KEY (eno) REFERENCES Emp(eno)  
                ON DELETE NO ACTION ON UPDATE CASCADE,  
    FOREIGN KEY (pno) REFERENCES Proj(pno)  
                ON DELETE NO ACTION ON UPDATE CASCADE  
);
```

Primary Key as a Foreign Key in its Own Relation



```
CREATE TABLE person
( idPerson int(10),
  idParent int(10) ,
  Gender char(5) DEFAULT NULL,
  PRIMARY KEY (idPerson),
  CONSTRAINT `SelfKey`
  FOREIGN KEY (idParent)
    REFERENCES person(idPerson)
    ON DELETE NO ACTION ON UPDATE NO ACTION )
```

ALTER TABLE



The ALTER TABLE command can be used to change an existing table. This is useful when the table already Contains data and you want to add or remove a column or constraint.

```
ALTER TABLE table_name ADD column_name datatype;
```

```
ALTER TABLE table_name DROP COLUMN column_name;
```

```
ALTER TABLE table_name MODIFY COLUMN column_name datatype;
```

```
ALTER TABLE table_name ADD CONSTRAINT constraint(column_name);
```

DROP TABLE



The command DROP TABLE is used to delete the table definition and all data from the database:

```
DROP TABLE table_name;
```

Indexes



Indexes are used to retrieve data from the database very fast. The users cannot see the indexes, they are just used to speed up searches/queries.

```
CREATE INDEX index_name ON table_name (column1);
```

```
ALTER TABLE table_name DROP INDEX index_name;
```

DML-

**Data Manipulation
Language**

SQL- SELECT using *

Querying with SQL is performed using a **SELECT** statement.

The "*" is used to select all attributes

Comparison operators can be used in Select: =, !=, >, <, >=, <=

SELECT * FROM SUPPLIER

SUPNR	SUPNAME	SUPADDRESS	SUPCITY	SUPSTATUS
21	Deliwines	240, Avenue of the Americas	New York	20
32	Best Wines	660, Market Street	San Francisco	90
37	Ad Fundum	82, Wacker Drive	Chicago	95
52	Spirits & co.	928, Strip	Las Vegas	NULL
68	The Wine Depot	132, Montgomery Street	San Francisco	10
69	Vinos del Mundo	4, Collins Avenue	Miami	92

SQL- SELECT using Column Names



SELECT SUPNR, SUPNAME FROM SUPPLIER

SUPNR	SUPNAME
21	Deliwines
32	Best Wines
37	Ad Fundum
52	Spirits & co.
68	The Wine Depot
69	Vinos del Mundo

SQL- SELECT using AS

SELECT SUPNR, PRODNR, DELIV_PERIOD/30 **AS**
MONTH_DELIV_PERIOD **FROM** SUPPLIES

SUPNR	PRODNR	MONTH_DELIV_PERIOD
21	0119	0.0333
21	0178	NULL
21	0289	0.0333
21	0327	0.2000
21	0347	0.0667
21	0384	0.0667
...

SQL- SELECT using WHERE

SELECT SUPNR, SUPNAME
FROM SUPPLIER **WHERE**
SUPCITY = 'San Francisco'

SUPNR	SUPNAME	SUPSTATUS
32	Best Wines	90
68	The Wine Depot	10

SELECT SUPNR, SUPNAME **FROM** SUPPLIER **WHERE**
SUPCITY = 'San Francisco' **AND** SUPSTATUS > 80

SQL- SELECT using WHERE & BETWEEN

```
SELECT SUPNR, SUPNAME, SUPSTATUS  
FROM SUPPLIER WHERE SUPSTATUS  
BETWEEN 70 AND 80
```

SUPNR	SUPNAME	SUPSTATUS
94	The Wine Crater	75

SQL- SELECT using WHERE & IN

SELECT SUPNR, SUPNAME, SUPSTATUS
FROM SUPPLIER **WHERE** SUPSTATUS
IN (70, 75, 80)

SUPNR	SUPNAME	SUPSTATUS
94	The Wine Crater	75

Other Cases for Select ... Where



Like

```
SELECT PRODNR, PRODNAME FROM PRODUCT  
      WHERE PRODNAME LIKE '%CHARD%'
```

IS NULL

```
SELECT SUPNR, SUPNAME, SUPSTATUS FROM SUPPLIER  
      WHERE SUPSTATUS IS NULL
```

Aggregate Functions



An aggregate function performs a calculation on a set of values and returns a single value

Examples: COUNT, SUM, AVG, VARIANCE, MIN, MAX, and STDEV

Aggregate Functions Examples

- How many movies in dataset?
- How many movies having names starting with **A**?
- How much would you earn if all movies get rented for 1 day?
- Calculate the average rent for movies? What are the highest and lowest rent values ?

123 film_id	release_year	123 rental_rate	123 rental_duration	123 length
1	2006	0.99	6	86
2	2006	4.99	3	48
3	2006	2.99	7	50
4	2006	2.99	5	117
5	2006	2.99	6	130
6	2006	2.99	3	169
7	2006	4.99	6	62
8	2006	4.99	6	54
9	2006	2.99	3	114
10	2006	4.99	6	63

```
select count(film_id) from film;
```

```
select count(film_id) from film where title like "A%";
```

```
select sum(rental_rate) from film;
```

```
select avg(rental_rate),max(rental_rate),min(rental_rate) from film;
```

SQL - Insert Statement

INSERT INTO PRODUCT

VALUES ('980', 'Chateau Angelus, Grand Clu Classé, 1960', 'red', 6)

INSERT INTO PRODUCT

(PRODNR, PRODNAME, PRODTYPE, AVAILABLE_QUANTITY)

VALUES ('980', 'Chateau Angelus, Grand Clu Classé, 1960', 'red', 6)

INSERT INTO PRODUCT

(PRODNR, PRODNAME, PRODTYPE, AVAILABLE_QUANTITY)

VALUES ('980', 'Chateau Angelus, Grand Clu Classé, 1960', 'red', 6),
('1000', 'Domaine de la, Bâtard Montrachet', 'Grand cru, 2010', 'white', 2)

SQL - Delete Statement

- **DELETE FROM PRODUCT WHERE PRODNR = '1000'**
- **DELETE FROM SUPPLIER WHERE SUPSTATUS IS NULL**
- **DELETE FROM SUPPLIES
WHERE PRODNR IN
(SELECT PRODNR
FROM PRODUCT
WHERE PRODNAME LIKE '%CHARD%')**

SQL - Update Statement

UPDATE PRODUCT

SET AVAILABLE_QUANTITY=26

WHERE PRODNR='0185'

UPDATE SUPPLIER SET SUPSTATUS = DEFAULT

UPDATE SUPPLIES

SET DELIV_PERIOD= DELIV_PERIOD+7

WHERE SUPNR IN

(SELECT SUPNR FROM SUPPLIER

WHERE SUPNAME = 'Deliwines')

QUESTIONS ???