CSc 484
Database Management Systems

Ken Gamradt

Spring 2024

Create Database

DDL

- Data Definition Language
  - Provides commands for
    - defining relation schemas
    - deleting relations
    - modifying relation schemas

Create a database

- SQL standard doesn't specify how databases are created
  - Each dialect generally has a different approach
- Database management tools provide functions on creating a database

```
create database databaseName;   -- PostgreSQL, MySQL, SQL Server support
```

- Check the manual references to learn additional options

Create a table

create table r
$(A_1 \quad D_1,$
$A_2 \quad D_2,$
$\ldots,$
$A_n \quad D_n,$
$\langle \text{integrity-constraint}_1 \rangle,$
$\ldots,$
$\langle \text{integrity-constraint}_k \rangle);$

```
create table instructor(
    ID              varchar(5),
    name            varchar(20) not null,
    dept_name       varchar(20),
    salary          numeric(8, 2),
    primary key (ID),
    foreign key (dept_name) references department);
```

Create a table – integrity constraints

- Ensure that changes made to the database by authorized users do not result in a loss of data consistency
  - Guard against accidental damage to the database
  - Are usually identified as part of the database schema design process and declared as part of the **CREATE TABLE** command
- Examples of integrity constraints are
  - An instructor name cannot be null
  - Not two instructors can have the same instructor ID
  - Every department name in the course relation must have a matching department name in the department relation
  - The budget of a department must be greater than $0.00

Create a table – integrity constraints

- Constraints on a single relation
  - **NOT NULL**
  - **UNIQUE**
  - **CHECK** (<predicate>)

Create a table – integrity constraints

- **NOT NULL**
  - **NULL** value is a member of all domains
  - **NOT NULL** constraints prohibits the insertion of a null value for a given attribute
    - Is an example of a **domain constraint**

Create a table – integrity constraints

- **UNIQUE**
  - No two tuples in the relation can be equal on all the listed attributes

unique *(columnName$_1$, columnName$_2$, …, columnName$_n$)*

- **UNIQUE** vs **PRIMARY KEY**
  - Attributes declared as unique are permitted to be null unless they have explicitly been declared to be **NOT NULL**
  - The **PRIMARY KEY** clause can be specified only once per table

Create a table – integrity constraints

- **CHECK** clause
  - ***check****(predicate)* -- specify a predicate that must be satisfied
- To ensure that attribute values satisfy specified conditions

```
create table section(
    course_id       varchar(8),
    sec_id          varchar(8),
    semester        varchar(6),
    year            numeric(4, 0),
    building        varchar(15),
    room_number     varchar(7),
    time_slot_id    varchar(4),
    primary key (course_id, sec_id, semester, year),
    check (semester in ('Fall', 'Winter', 'Spring','Summer')));
```

-- semester must be one of those specified

Create a table – integrity constraints

- **CHECK** clause may appear on its own, or as part of the declaration of an attribute

```
create table section(
    course_id       varchar(8),
    sec_id          varchar(8),
    semester        varchar(6) check (semester in
                                ('Fall', 'Winter', 'Spring','Summer')),
    year            numeric(4, 0),
    building        varchar(15),
    room_number     varchar(7),
    time_slot_id    varchar(4),
    primary key (course_id, sec_id, semester, year));
```

Create a table – integrity constraints

- Another example of **CHECK**

```
create table department(
    dept_name       varchar(20),
    building        varchar(15),
    budget          numeric(12, 2) check (budget > 0),
    primary key (dept_name));
```

Create a table – integrity constraints

- Referential integrity: **FOREIGN KEY** clause
  - A value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relation

```
create table instructor(
    ID              varchar(5),
    name            varchar(20) not null,
    dept_name       varchar(20),
    salary          numeric(8, 2) check (salary > 29000),
    primary key (ID),
    foreign key (dept_name) references department);
```

Create a table – integrity constraints

- **FOREIGN KEY** clause
  - By default, a foreign key references the primary-key attributes of the referenced table
  - The foreign key must reference a compatible set of attributes
    - The number of attributes must be the same
    - The data types of corresponding attributes must be compatible

```
FOREIGN KEY (dept_name) REFERENCES department;
-- SQL standard supports both formats
FOREIGN KEY (dept_name) REFERENCES department (dept_name);
-- explicitly specifies the the referenced attributes
-- must be declared as a superkey of the relation – primary key or unique
```

Create a table – integrity constraints

```sql
create table department(
    dept_name       varchar(20),
    building        varchar(15),
    budget          numeric(12, 2) check (budget > 0),
    primary key (dept_name));

create table instructor(
    ID              varchar(5),
    name            varchar(20) not null,
    dept_name       varchar(20),
    salary          numeric(8, 2) check (salary > 29000),
    primary key (ID),
    foreign key (dept_name) references department);

delete from department
    where dept_name = 'Physics';
-- rejected because of the
-- foreign-key constraint
```

	dept_name	building	budget
1	Biology	Watson	90000.00
2	Comp. Sci.	Taylor	100000.00
3	Elec. Eng.	Taylor	85000.00
4	Finance	Painter	120000.00
5	History	Painter	50000.00
6	Music	Packard	80000.00
7	Physics	Watson	70000.00

	ID	name	dept_name	salary
1	10101	Srinivasan	Comp. Sci.	65000.00
2	12121	Wu	Finance	90000.00
3	15151	Mozart	Music	40000.00
4	22222	Einstein	Physics	95000.00
5	32343	El Said	History	60000.00
6	33456	Gold	Physics	87000.00
7	45565	Katz	Comp. Sci.	75000.00
8	58583	Califieri	History	62000.00

Create a table – integrity constraints

- **ON DELETE CASCADE**
  - Delete department information from department relation

```
create table instructor(
    ID              varchar(5),
    name            varchar(20) not null,
    dept_name       varchar(20),
    salary          numeric(8, 2) check (salary > 29000),
    primary key (ID),
    foreign key (dept_name) references department (dept_name)
        on delete cascade);
```

- Specify if delete action on the referenced relation violates the constraints
- The system **also deletes the tuples** in the referencing relation to restore the constraint

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000.00
12121	Wu	Finance	90000.00
15151	Mozart	Music	40000.00
22222	Einstein	Physics	95000.00
32343	El Said	History	60000.00
33456	Gold	Physics	87000.00
45565	Katz	Comp. Sci.	75000.00
58583	Califieri	History	62000.00
76543	Singh	Finance	80000.00
76766	Crick	Biology	72000.00
83821	Brandt	Comp. Sci.	92000.00
98345	Kim	Elec. Eng.	80000.00

delete from department
where dept_name = 'Physics';

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000.00
12121	Wu	Finance	90000.00
15151	Mozart	Music	40000.00
32343	El Said	History	60000.00
45565	Katz	Comp. Sci.	75000.00
58583	Califieri	History	62000.00
76543	Singh	Finance	80000.00
76766	Crick	Biology	72000.00
83821	Brandt	Comp. Sci.	92000.00
98345	Kim	Elec. Eng.	80000.00

Create a table – integrity constraints

- **ON UPDATE CASCADE**
  - Update a department's name in department relation

```
create table instructor(
    ID              varchar(5),
    name            varchar(20) not null,
    dept_name       varchar(20),
    salary          numeric(8, 2) check (salary > 29000),
    primary key (ID),
    foreign key (dept_name) references department (dept_name)
        on update cascade);
```

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000.00
12121	Wu	Finance	90000.00
15151	Mozart	Music	40000.00
22222	Einstein	Physics	95000.00
32343	El Said	History	60000.00
33456	Gold	Physics	87000.00
45565	Katz	Comp. Sci.	75000.00
58583	Califieri	History	62000.00
76543	Singh	Finance	80000.00
76766	Crick	Biology	72000.00
83821	Brandt	Comp. Sci.	92000.00
98345	Kim	Elec. Eng.	80000.00

```
update department
  set dept_name = 'CS'
  where dept_name = 'Comp. Sci.';
```

ID	name	dept_name	salary
10101	Srinivasan	CS	65000.00
12121	Wu	Finance	90000.00
15151	Mozart	Music	40000.00
22222	Einstein	Physics	95000.00
32343	El Said	History	60000.00
33456	Gold	Physics	87000.00
45565	Katz	CS	75000.00
58583	Califieri	History	62000.00
76543	Singh	Finance	80000.00
76766	Crick	Biology	72000.00
83821	Brandt	CS	92000.00
98345	Kim	Elec. Eng.	80000.00

- Specify if an update action on the referenced relation violates the constraints
- The system **also updates the tuples** in the referencing relation to restore the constraint

Create a table – integrity constraints

- **ON DELETE SET NULL**
  - Delete a department information in department relation

```
create table instructor(
    ID              varchar(5),
    name            varchar(20) not null,
    dept_name       varchar(20),
    salary          numeric(8, 2) check (salary > 29000),
    primary key (ID),
    foreign key (dept_name) references department (dept_name)
        on delete set null);
```

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000.00
12121	Wu	Finance	90000.00
15151	Mozart	Music	40000.00
22222	Einstein	Physics	95000.00
32343	El Said	History	60000.00
33456	Gold	Physics	87000.00
45565	Katz	Comp. Sci.	75000.00
58583	Califieri	History	62000.00
76543	Singh	Finance	80000.00
76766	Crick	Biology	72000.00
83821	Brandt	Comp. Sci.	92000.00
98345	Kim	Elec. Eng.	80000.00

delete from department
where dept_name = 'Physics';

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000.00
12121	Wu	Finance	90000.00
15151	Mozart	Music	40000.00
22222	Einstein	NULL	95000.00
32343	El Said	History	60000.00
33456	Gold	NULL	87000.00
45565	Katz	Comp. Sci.	75000.00
58583	Califieri	History	62000.00
76543	Singh	Finance	80000.00
76766	Crick	Biology	72000.00
83821	Brandt	Comp. Sci.	92000.00
98345	Kim	Elec. Eng.	80000.00

- Specify if a delete action on the referenced relation violates the constraints
- The system **set NULL value to the tuples** in the referencing relation to restore the constraint

Create a table – default values

- SQL allows a default value to be specified for an attribute

```
create table student(
    ID              varchar(5),
    name            varchar(20) not null,
    dept_name       varchar(20),
    tot_cred        numeric(3, 0) default 0,
    primary key (ID));
```

Removing a table – Drop Table

```
drop table instructor;  -- Not able to delete a table as a referenced relation
                        -- with a foreign key constraint
```

Changing a table definition – Alter Table

Specify the table to be altered

Add a new column to a table

Drop a column from a table

Add a new table constraint

Drop a table constraint

Set a default for a column

Drop a default for a column

```
ALTER TABLE TableName
[ADD [COLUMN] columnName dataType [NOT NULL] [UNIQUE]
[DEFAULT defaultOption] [CHECK (searchCondition)]]
[DROP [COLUMN] columnName [RESTRICT | CASCADE]]
[ADD [CONSTRAINT [ConstraintName]] tableConstraintDefinition]
[DROP CONSTRAINT ConstraintName [RESTRICT | CASCADE]]
[ALTER [COLUMN] SET DEFAULT defaultOption]
[ALTER [COLUMN] DROP DEFAULT]
```

Changing a table definition – Alter Table

- Add a new column to a table
- E.g., add a new attribute to instructor relation for recording addresses

```
alter table instructor
    add address varchar(50);
```

	ID	name	dept_name	salary	address
1	10101	Srinivasan	Comp. Sci.	65000.00	NULL
2	12121	Wu	Finance	90000.00	NULL
3	15151	Mozart	Music	40000.00	NULL
4	22222	Einstein	Physics	95000.00	NULL
5	32343	El Said	History	60000.00	NULL
6	33456	Gold	Physics	87000.00	NULL
7	45565	Katz	Comp. Sci.	75000.00	NULL
8	58583	Califieri	History	62000.00	NULL
9	76543	Singh	Finance	80000.00	NULL

Changing a table definition – Alter Table

- Add a new column to the table
- E.g., add an attribute for recording the gender of instructors, set the default value to be 'F', also make sure the values for gender can only be 'F' or 'M'

```
alter table instructor
    add gender varchar(2) default 'F' check(gender in ('F', 'M'));
```

	ID	name	dept_name	salary	gender
1	10101	Srinivasan	Comp. Sci.	65000.00	NULL
2	12121	Wu	Finance	90000.00	NULL
3	15151	Mozart	Music	40000.00	NULL
4	22222	Einstein	Physics	95000.00	NULL
5	32343	El Said	History	60000.00	NULL
6	33456	Gold	Physics	87000.00	NULL
7	45565	Katz	Comp. Sci.	75000.00	NULL

Changing a table definition – Alter Table

- Drop a column from a table

```
alter table instructor
    drop column address;    -- SQL Server

alter table instructor
    drop address;           -- PostgreSQL
```

Changing a table definition – Alter Table

- PostgreSQL

```
alter table instructor
    add gender varchar(2) default 'F' check(gender in ('F', 'M'));

alter table instructor
    drop address;
```

	id [PK] character varying (5)	name character varying (20)	dept_name character varying (20)	salary numeric (8,2)	gender character varying (2)
1	10101	Srinivasan	Comp. Sci.	65000.00	F
2	12121	Wu	Finance	90000.00	F
3	15151	Mozart	Music	40000.00	F
4	22222	Einstein	Physics	95000.00	F
5	32343	El Said	History	60000.00	F

Changing a table definition – Alter Table

- Drop a column from a table

```
alter table instructor
    drop column gender;
```

```
Msg 5074, Level 16, State 1, Line 63
The object 'DF__instructo__gende__6477ECF3' is dependent on column 'gender'.
Msg 5074, Level 16, State 1, Line 63
The object 'CK__instructo__gende__656C112C' is dependent on column 'gender'.
Msg 4922, Level 16, State 9, Line 63
ALTER TABLE DROP COLUMN gender failed because one or more objects access this column.
```

Changing a table definition – Alter Table

- Drop a table constraint

```
alter table instructor
    drop constraint DF__instructo__gende__6477ECF3;

-- constraint names

alter table instructor
    drop constraint CK__instructo__gende__656C112C;
```

Acknowledgements

- WIKIPEDIA
  - https://en.wikipedia.org/wiki/SQL