Lecture 12

The Structured Query Language (SQL)

Week 6

SQL: Structured Query Language

- Both a Data Definition Language (DDL) and Data Manipulation Language (DML)
- We study SQL2: Standard:
 - ANSI X3,135-1992 (American / US)
 - ISO 9075:1992 (International)

© 2009 Griffith Universit

Data Definition in SQL

• Defines a schema and the tables belonging to it

2009 Griffith Universit

Some Remarks

- · Relation is called a Table
- Attribute is called a column
- Tuple is called a *row* (we use these interchangeably)
- A Table may contain identical rows (i.e. not a <u>set</u> of tuples, but a *bag* of tuples)
- We can declare if we want unique tuples

© 2009 Griffith Universi

Catalog

- A Catalog is the data dictionary describing all databases maintained by-, or known to a database management system
- SQL2: Each database has a *Schema*CREATE SCHEMA company AUTHORIZATION smith

 Owner of schema
- Oracle SQL*Plus: one schema per user CREATE user smith identified by <passwd>

User and schema name

2009 Griffith Universit

Schema

 A Schema consists of tables and associated constraints
 Table of schema 'company' (SQL2)

CREATE TABLE company.employee ...

CREATE TABLE employee

Table of currently 'open' schema (and Oracle SQL*Plus)

2009 Griffith University

Attributes (columns)

• Attributes (columns) have a data type

```
CHAR, CHAR(n), VARCHAR(n), INT, BIT(n), DATE, TIME, FLOAT, DOUBLE, etc...
```

• Also, named value domains can be defined:

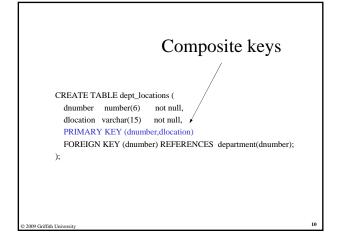
```
CREATE DOMAIN ssn_type AS CHAR (9)
```

(increases readability and improves maintainability)

2009 Griffith Univers

```
Example: create employee
      CREATE TABLE employee (
                  varchar(20)
                                 not null,
         name
                  char(9)
                                 not null,
        bdate
                  date.
         address
                  varchar(40),
                  char(1),
                                                 constraints
         salary
                  number(10),
         superssn
                  char(9).
         dno
                  number(6),
         PRIMARY KEY (ssn)
                               value domains
attributes
```

Constraints CREATE TABLE employee (varchar(20) not null, char(9) ssn not null, bdate date, address varchar(40), This constraints has no name char(1), salary number(10), char(9). superssn This constraint has a name dno number(6), PRIMARY KEY (ssn); CONSTRAINT employee_dn FOREIGN KEY (dno) REFERENCES department(dnumber);



```
Other candidate keys

CREATE TABLE project (
    pname varchar(15) not pull,
    pnumber number(6) not null, /* primary key */
    plocation varchar(15),
    dnum number(6) not null, /* references department */
    PRIMARY KEY (pnumber),
    UNIQUE (pname)
);
```

```
Special Problem with foreign keys:
what to do when database changes?

CREATE TABLE employee (
ssn char(9) not null,
.........

superssn char(9),
dno number(6),

PRIMARY KEY (ssn);

CONSTRAINT employee_superssn
FOREIGN KEY (superssn) REFERENCES employee(ssn);

CONSTRAINT employee_dno
FOREIGN KEY (dno) REFERENCES department(dnumber);
);
```

Special Problem with foreign keys: what to do when database changes? CREATE TABLE employee (not null. When the referenced ssn char(9) tuple is deleted, the referring attribute will be set to null char(9), superssn When the referred attribute dno number(6), changes all references to it also change (not in Oracle 8, but in SQL92) PRIMARY KEY (ssn): CONSTRAINT employee_superssn FOREIGN KEY (superson) REFERENCES employee(ssn) ON DELETE set mull ON UPDATE cascade; CONSTRAINT employee_dno FOREIGN KEY (dno) REFERENCES department(dnumber);

