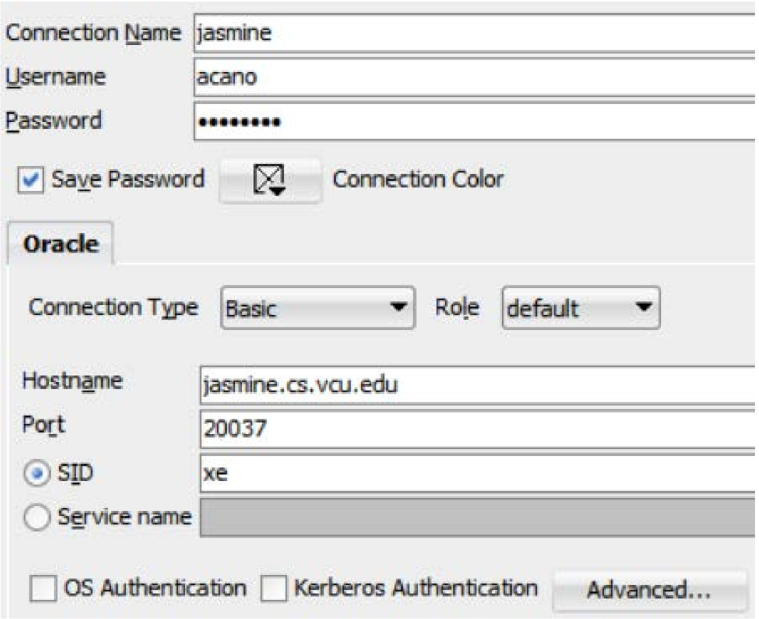
CMSC 508
Database Theory

Introduction to SQL (I)

Dr. Alberto Cano
Assistant Professor
Department of Computer Science

Chapter 3 from Database System Concepts, 6th Ed. by Silberschatz, Korth, Sudarshan, 2011
Chapter 5 from Database Management Systems, 3rd Ed. by Ramakrishnan, Gehrke, 2003

- Oracle SQL Developer
  - Download from [Oracle website]
  - Connection information:
    - Username: uname (as in uname@vcu.edu)
    - Password: V0123456 (your V#, NOT your VCU password)
    - Hostname: jasmine.cs.vcu.edu
    - Port: 20037
    - SID: xe
  - Use VCU [VPN] to connect from home anytime!



Connection Name	jasmine
Username	acano
Password	•••••••••

☑ Save Password Connection Color

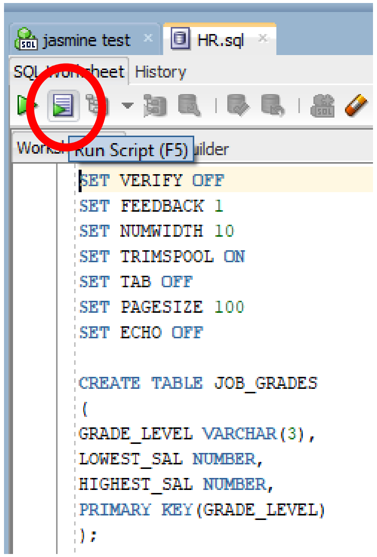
Oracle

Connection Type: Basic Role: default

Hostname	jasmine.cs.vcu.edu
Port	20037
● SID	xe
○ Service name	

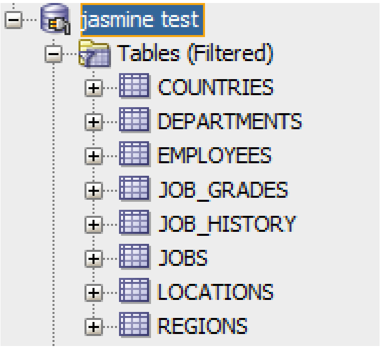
□ OS Authentication □ Kerberos Authentication

- Human Resources Database
  - Download the Data Model definition (HR.sql) This file includes all the DDL instructions to create the tables, triggers, views, etc of the DB
  - Open the HR.sql file using Oracle SQL Developer
  - Execute the script **ONLY ONCE** by clicking on the run script icon. This will create your database, please be patient, there is much stuff to do
  - Finally refresh the connection (refresh icon) and you will see the tables created

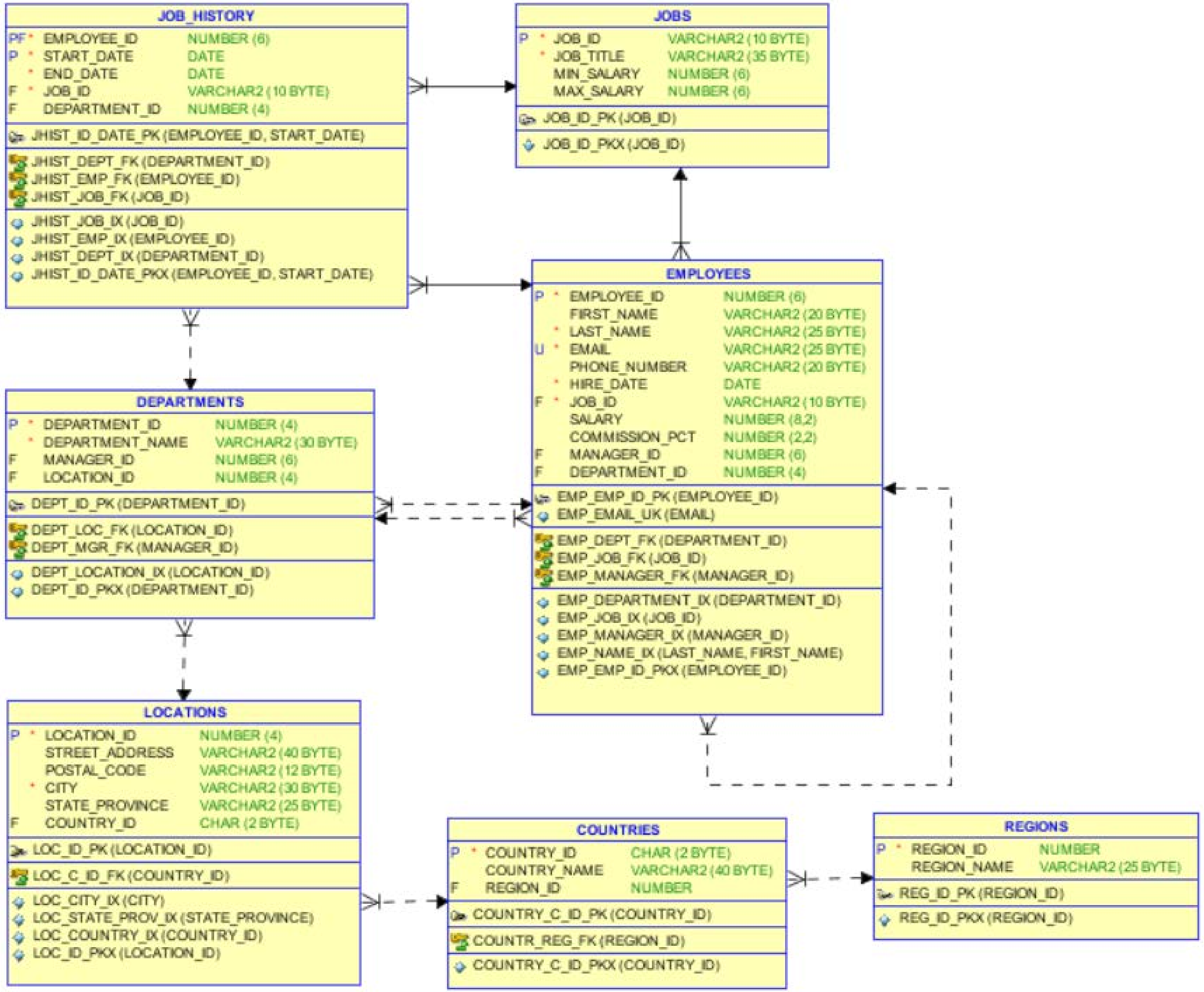


```
SET VERIFY OFF
SET FEEDBACK 1
SET NUMWIDTH 10
SET TRIMSPOOL ON
SET TAB OFF
SET PAGESIZE 100
SET ECHO OFF

CREATE TABLE JOB_GRADES
(
GRADE_LEVEL VARCHAR(3),
LOWEST_SAL NUMBER,
HIGHEST_SAL NUMBER,
PRIMARY KEY(GRADE_LEVEL)
);
```



- Human Resources Database (Relational diagram with PK/FK keys)



JOB_HISTORY

PF	*	EMPLOYEE_ID	NUMBER (6)
P	*	START_DATE	DATE
	*	END_DATE	DATE
F	*	JOB_ID	VARCHAR2 (10 BYTE)
F		DEPARTMENT_ID	NUMBER (4)

JHIST_ID_DATE_PK (EMPLOYEE_ID, START_DATE)

JHIST_DEPT_FK (DEPARTMENT_ID)
JHIST_EMP_FK (EMPLOYEE_ID)
JHIST_JOB_FK (JOB_ID)

JHIST_JOB_IX (JOB_ID)
JHIST_EMP_IX (EMPLOYEE_ID)
JHIST_DEPT_IX (DEPARTMENT_ID)
JHIST_ID_DATE_PKX (EMPLOYEE_ID, START_DATE)

JOBS

P	*	JOB_ID	VARCHAR2 (10 BYTE)
	*	JOB_TITLE	VARCHAR2 (35 BYTE)
		MIN_SALARY	NUMBER (6)
		MAX_SALARY	NUMBER (6)

JOB_ID_PK (JOB_ID)

JOB_ID_PKX (JOB_ID)

EMPLOYEES

P	*	EMPLOYEE_ID	NUMBER (6)
		FIRST_NAME	VARCHAR2 (20 BYTE)
	*	LAST_NAME	VARCHAR2 (25 BYTE)
U	*	EMAIL	VARCHAR2 (25 BYTE)
		PHONE_NUMBER	VARCHAR2 (20 BYTE)
	*	HIRE_DATE	DATE
F	*	JOB_ID	VARCHAR2 (10 BYTE)
		SALARY	NUMBER (8,2)
		COMMISSION_PCT	NUMBER (2,2)
F		MANAGER_ID	NUMBER (6)
F		DEPARTMENT_ID	NUMBER (4)

EMP_EMP_ID_PK (EMPLOYEE_ID)
EMP_EMAIL_UK (EMAIL)

EMP_DEPT_FK (DEPARTMENT_ID)
EMP_JOB_FK (JOB_ID)
EMP_MANAGER_FK (MANAGER_ID)

EMP_DEPARTMENT_IX (DEPARTMENT_ID)
EMP_JOB_IX (JOB_ID)
EMP_MANAGER_IX (MANAGER_ID)
EMP_NAME_IX (LAST_NAME, FIRST_NAME)
EMP_EMP_ID_PKX (EMPLOYEE_ID)

DEPARTMENTS

P	*	DEPARTMENT_ID	NUMBER (4)
	*	DEPARTMENT_NAME	VARCHAR2 (30 BYTE)
F		MANAGER_ID	NUMBER (6)
F		LOCATION_ID	NUMBER (4)

DEPT_ID_PK (DEPARTMENT_ID)

DEPT_LOC_FK (LOCATION_ID)
DEPT_MGR_FK (MANAGER_ID)

DEPT_LOCATION_IX (LOCATION_ID)
DEPT_ID_PKX (DEPARTMENT_ID)

LOCATIONS

P	*	LOCATION_ID	NUMBER (4)
		STREET_ADDRESS	VARCHAR2 (40 BYTE)
		POSTAL_CODE	VARCHAR2 (12 BYTE)
	*	CITY	VARCHAR2 (30 BYTE)
		STATE_PROVINCE	VARCHAR2 (25 BYTE)
F		COUNTRY_ID	CHAR (2 BYTE)

LOC_ID_PK (LOCATION_ID)

LOC_C_ID_FK (COUNTRY_ID)

LOC_CITY_IX (CITY)
LOC_STATE_PROV_IX (STATE_PROVINCE)
LOC_COUNTRY_IX (COUNTRY_ID)
LOC_ID_PKX (LOCATION_ID)

COUNTRIES

P	*	COUNTRY_ID	CHAR (2 BYTE)
		COUNTRY_NAME	VARCHAR2 (40 BYTE)
F		REGION_ID	NUMBER

COUNTRY_C_ID_PK (COUNTRY_ID)

COUNTR_REG_FK (REGION_ID)

COUNTRY_C_ID_PKX (COUNTRY_ID)

REGIONS

P	*	REGION_ID	NUMBER
		REGION_NAME	VARCHAR2 (25 BYTE)

REG_ID_PK (REGION_ID)

REG_ID_PKX (REGION_ID)

- Human Resources Database
  - The HR.sql script can be run only once (cannot duplicate tables, etc)
  - Should you need **to rebuild the original database**, drop the current content **run the following commands and reload the script HR.sql**

```
DROP TABLE COUNTRIES cascade constraints;
DROP TABLE DEPARTMENTS cascade constraints;
DROP TABLE EMPLOYEES cascade constraints;
DROP TABLE JOBS cascade constraints;
DROP TABLE JOB_HISTORY cascade constraints;
DROP TABLE JOB_GRADES cascade constraints;
DROP TABLE LOCATIONS cascade constraints;
DROP TABLE REGIONS cascade constraints;
DROP SEQUENCE DEPARTMENTS_SEQ;
DROP SEQUENCE EMPLOYEES_SEQ;
DROP SEQUENCE LOCATIONS_SEQ;
DROP VIEW EMP_DETAILS_VIEW;
DROP PROCEDURE ADD_JOB_HISTORY;
DROP PROCEDURE SECURE_DML;
```

- Data Definition Language
  - The SQL data-definition language (DDL) allows the specification of information about relations, including:
    - The schema for each relation
    - The domain of values associated with each attribute
    - Integrity constraints
    - Other information such as:
      - The set of indices to be maintained for each relation
      - Security and authorization information for each relation
      - The physical storage structure of each relation on disk

- Oracle Data Types
  - CHAR(size)

  Stores fixed-length character strings, must specify a string length (in bytes or characters) between 1 and 2000 bytes. The default is 1 byte

  - VARCHAR(size) and VARCHAR2(size)

  Stores variable-length character strings, you specify a maximum string length (in bytes or characters) between 1 and 4000 bytes

  VARCHAR supports distinction between NULL and empty string

  VARCHAR2 does not distinguish between a NULL and empty string

  If you rely on empty string and NULL being the same thing, you should use VARCHAR2

- Oracle Data Types
  - NUMBER

    Stores fixed and floating-point numbers. Numbers of virtually any magnitude can be stored and are guaranteed portable among different systems operating, up to 38 digits of precision

    Optionally, you can also specify a precision p (total number of digits) and scale s (number of digits to the right of the decimal point) as NUMBER(p,s), and also NUMBER(*,s) for any unknown precision
  - DATE

    Stores point-in-time values (dates and times), i.e., the year (including the century), the month, the day, the hours, the minutes, and the seconds (after midnight)

- Oracle Data Types
  - BLOB

    Stores unstructured binary data up to 128 terabytes. Enables to store and manipulate large blocks of unstructured data (such as text, graphic images, video, and sound waveforms)

  - BFILE

    Stores unstructured binary data in operating-system files outside the database. Uses a file locator that points to an external file containing the data. The amount of BFILE data that can be stored is limited by the operating system. BFILEs are read only

- CREATE TABLE
  - An SQL relation is defined using the create table command:

    create table *r* (A_1 D_1, A_2 D_2, ..., A_n D_n,
    (integrity-constraint$_1$),
    ...,
    (integrity-constraint$_k$))

    - *r* is the name of the relation
    - A_i is an attribute name in the schema of relation *r*
    - D_i is the data type of values in the domain of attribute A_i

Example:

```
create table instructor (
    ID           char(5),
    name         varchar(20),
    dept_name    varchar(20),
    salary       number(8,2));
```

- Integrity Constraints in Create Table
  - not null
  - primary key (A_n, ..., A_m)
  - foreign key (A_j, ..., A_k) references r

Example:

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

primary key declaration on an attribute ensures **not null**

- Integrity Constraints in Create Table

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

SQL Error: ORA-00955: name is already used by an existing object
Instructor table already exists!

Drop current table and create again

```
drop table instructor;
```

- Integrity Constraints in Create Table

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

SQL Error: ORA-00942: table or view does not exist

The foreign key references table department which does not exist

```
create table department (
    dept_name varchar (20),   building varchar (15),
    budget number(12,2),      primary key (dept_name));
```

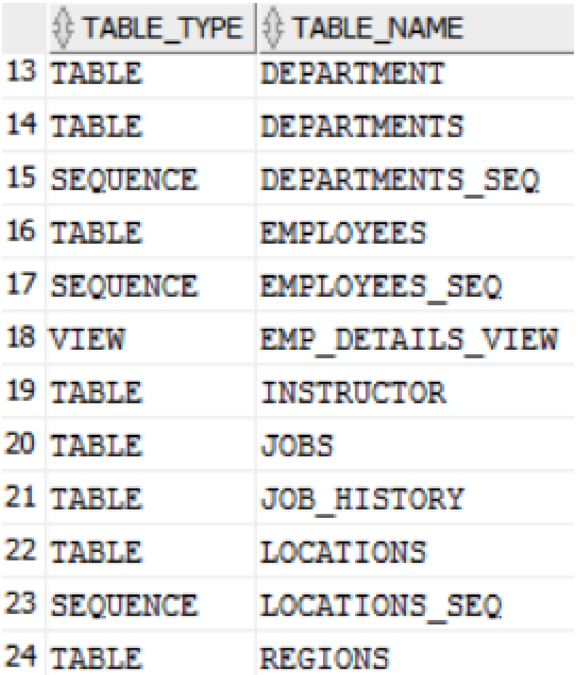
- DESCRIBE
  - Lists the column definitions for the specified table or view, or the specifications for the specified function or procedure

describe *instructor*;

```
Name          Null        Type
---------     --------    ------------
ID            NOT NULL    CHAR(5)
NAME          NOT NULL    VARCHAR2(20)
DEPT_NAME                 VARCHAR2(20)
SALARY                    NUMBER(8,2)
```

- USER_CATALOG
  - Lists all objects owned by the user connected to Oracle

select * from *user_catalog*;



	TABLE_TYPE	TABLE_NAME
13	TABLE	DEPARTMENT
14	TABLE	DEPARTMENTS
15	SEQUENCE	DEPARTMENTS_SEQ
16	TABLE	EMPLOYEES
17	SEQUENCE	EMPLOYEES_SEQ
18	VIEW	EMP_DETAILS_VIEW
19	TABLE	INSTRUCTOR
20	TABLE	JOBS
21	TABLE	JOB_HISTORY
22	TABLE	LOCATIONS
23	SEQUENCE	LOCATIONS_SEQ
24	TABLE	REGIONS

- INSERT
  - Adds new rows of data to a table

insert into r **values** (V_1, V_2, …, V_n); *where V_i corresponds to A_i*

insert into *instructor* **values** *('10211', 'Smith', 'Biology', 66000);*

SQL Error: ORA-02291: integrity constraint violated - parent key not found

Instructor *dept_name* references *department* table and there is no department named ‘Biology’. Referential integrity violation.

insert into r (A_1, A_2, A_3); **values** (V_1, V_2, V_3); *explicit attribute-value*

insert into *department (dept_name, building)*
values *('Biology', 'West Hall');*

This creates the Biology department, but budget is not provided

- UPDATE
  - Modifies the content of one or more rows of a table

```
update department
set budget = 25000
where dept_name = 'Biology';            ← do not forget where clause
```

```
update instructor
set salary = salary * 1.05
where name = 'Smith';
```

Increases 5% the salary of **ALL** instructors named Smith, probably not a good idea. ALWAYS use PK identifiers to limit **where** clauses.

```
update instructor
set salary = salary * 1.07;      ← Update all instructors' salary 7%!
```

- UPDATE
  - Case statement for conditional updates

```
update instructor
set salary = case
    when salary <= 100000
        then salary * 1.05
        else salary * 1.03
end;
```

- DELETE
  - Removes rows of data from a table

delete from *department;* ← *removes all departments*

SQL Error: ORA-02292: integrity constraint violated - child record found

Removes **ALL** rows from department. However, there's a row in the instructor table referencing the Biology department. Therefore, we cannot remove the Biology department without violating the referential integrity.

delete from *instructor* **where** name = *'Smith';*

Removes from instructor ALL instructors named Smith, probably not a good idea. ALWAYS use PK identifiers in the **where** clause.

- ALTER TABLE
  - Modifies the definition of a database table

alter table *r* **add** *A D*

where *A* is the name of the attribute to be added to relation *r* and *D* is the domain of *A.* All exiting tuples in the relation are assigned *null* as the value for the new attribute.

alter table *r* **drop column** *A*

where *A* is the name of an attribute of relation *r*

Dropping of attributes not supported by many databases.

alter table *instructor* **add** *email* **varchar**(25)*;*

alter table *instructor* **drop column** *email*;

- TRUNCATE TABLE
  - Removes ALL rows from a table. Can be more efficient than dropping and re-creating a table. Dropping and re-creating a table requires to recreate the indexes, integrity constraints, and triggers on the table. Truncating has none of these effects.

truncate table *department;*

If you didn’t delete Smith before, he’s still enforcing referential integrity

SQL Error: ORA-02266: unique/primary keys in table referenced by enabled foreign keys

**Cause: An attempt was made to truncate a table with unique or primary keys referenced by foreign keys enabled in another table.*

- DROP TABLE
  - Removes the table and all its data from the database entirely

drop table *instructor;*

drop table *department;*

- SYSDATE
  - Obtains the current date (timestamp)

select *sysdate* **from** *dual***;**

insert into *tablename* (*current_time*) **values** (*sysdate*);

- Transaction Control
  - COMMIT makes permanent any database changes made during the current transaction
  - ROLLBACK ends the current transaction and undoes any changes made since the transaction began
  - SAVEPOINT marks the current point in the processing of a transaction

```
savepoint savepointname1;
update employees set salary = 7000 where last_name = 'Banda';
rollback to savepoint savepointname1;
update employees set salary = 11000 where last_name = 'Greene';
commit;
```

- Exercises to practice

1. Create a savepoint (important!)
2. Create a table to store the set of countries visited by the employees
3. Insert some rows to the previous table
4. Insert a new employee with all the required details
5. Insert a row into departments table with manager ID 120 and location ID in any location ID for city Tokyo (see locations table)
6. Change salary of employee 115 to 8000 if the existing salary is less than 6000 and higher than 5000
7. Change job ID of employee 110 to IT_PROG if the employee belongs to department 10
8. Delete department 20
9. Restore the database to the savepoint (make sure to undo all changes!)

CMSC 508
Database Theory

Introduction to SQL (I)

Dr. Alberto Cano
Assistant Professor
Department of Computer Science

Chapter 3 from Database System Concepts, 6th Ed. by Silberschatz, Korth, Sudarshan, 2011
Chapter 5 from Database Management Systems, 3rd Ed. by Ramakrishnan, Gehrke, 2003