讲师: collen7788@126.com

使用DDL语句创建和管理表

本章目标

1. 描述主要的数据库对象。
2. 创建表。
3. 描述各种数据类型。
4. 修改表的定义。
5. 删除，重命名和清空表。

常见的数据库对象

对象	描述
表	基本的数据存储集合，由行和列组成。
视图	从表中抽出的逻辑上相关的数据集合。
序列	提供有规律的数值。
索引	提高查询的效率
同义词	给对象起别名

命名规则

- **表名和列名**
  - 必须以字母开头
  - 必须在 1–30 个字符之间
  - 必须只能包含 A–Z, a–z, 0–9, _, $, 和 #
  - 必须不能和用户定义的其他对象重名
  - 必须不能是Oracle 的保留字
  - Oracle默认存储是都存为大写
  - 数据库名只能是1~8位，datalink可以是128位，和其他一些特殊字符

CREATE TABLE 语句

- **必须具备:**
  - CREATE TABLE权限
  - 存储空间

```
CREATE TABLE [schema.]table
            (column datatype [DEFAULT expr][, ...]);
```

- **必须指定:**
  - 表名
  - 列名, 数据类型, 数据类型的大小

Default值

- 执行**insert**操作时，可以为其指定默认值

```
... hire_date DATE DEFAULT SYSDATE, ...
```

- 值、表达式和**SQL**语句都可以作为默认值
- 其他的列名或者是伪列都是非法的
- 默认值的类型必须和该列的类型一致

```
CREATE TABLE hire_dates
       (id          NUMBER(8),
        hire_date DATE DEFAULT SYSDATE);
Table created.
```

创建表

- 语法

```
CREATE TABLE dept
        (deptno  NUMBER(2),
         dname   VARCHAR2(14),
         loc     VARCHAR2(13));
Table created.
```

- 确认

```
DESCRIBE dept
```

Name	Null?	Type
DEPTNO		NUMBER(2)
DNAME		VARCHAR2(14)
LOC		VARCHAR2(13)

数据类型

数据类型	描述
VARCHAR2(*size*)	可变长字符数据
CHAR(*size*)	定长字符数据
NUMBER(*p*,*s*)	可变长数值数据
DATE	日期型数据
LONG	可变长字符数据，最大可达到2G
CLOB	字符数据，最大可达到4G
RAW and LONG RAW	原始的二进制数据
BLOB	二进制数据，最大可达到4G
BFILE	存储外部文件的二进制数据，最大可达到4G
ROWID	行地址

使用子查询创建表

- 使用 **AS *subquery*** 选项，将创建表和插入数据结合起来

```
CREATE TABLE table
        [(column, column...)]
AS subquery;
```

- 指定的列和子查询中的列要一一对应
- 通过列名和默认值定义列

使用子查询创建表举例

```
CREATE TABLE  dept80
  AS
    SELECT  employee_id, last_name,
            salary*12 ANNSAL,
            hire_date
    FROM    employees
    WHERE   department_id = 80;
Table created.
```

```
 DESCRIBE dept80
```

Name	Null?	Type
EMPLOYEE_ID		NUMBER(6)
LAST_NAME	NOT NULL	VARCHAR2(25)
ANNSAL		NUMBER
HIRE_DATE	NOT NULL	DATE

ALTER TABLE 语句

- 使用 **ALTER TABLE** 语句可以：
  - 追加新的列
  - 修改现有的列
  - 删除一个列

ALTER TABLE 语句

- 使用 **ALTER TABLE** 语句追加，修改，或删除列的语法.

```
ALTER TABLE table
ADD         (column datatype [DEFAULT expr]
            [, column datatype]...);
```

```
ALTER TABLE table
MODIFY      (column datatype [DEFAULT expr]
            [, column datatype]...);
```

```
ALTER TABLE table
DROP  column    (column);
```

```
ALTER TABLE table_name rename column old_column_name
to new_column_name
```

追加一个新列

DEPT80

EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE
149	Zlotkey	126000	29-JAN-00
174	Abel	132000	11-MAY-96
176	Taylor	103200	24-MAR-98

DEPT80

EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE	JOB_ID
149	Zlotkey	126000	29-JAN-00	
174	Abel	132000	11-MAY-96	
176	Taylor	103200	24-MAR-98	

追加一个新列

- 使用 **ADD** 子句追加一个新列

```
ALTER TABLE dept80
ADD       (job_id VARCHAR2(9));
Table altered.
```

- 新列是表中的最后一列

EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE	JOB_ID
149	Zlotkey	126000	29-JAN-00	
174	Abel	132000	11-MAY-96	
176	Taylor	103200	24-MAR-98	

修改一个列

- 可以修改列的数据类型，尺寸，和默认值

```
ALTER TABLE   dept80
MODIFY        (last_name VARCHAR2(30));
Table altered.
```

- 对默认值的修改只影响今后对表的修改

删除一个列

- 使用 **DROP COLUMN** 子句删除不再需要的列.

```
ALTER TABLE  dept80
DROP COLUMN  job_id;
Table altered.
```

删除表

- 数据和结构都被删除
- 所有正在运行的相关事物被提交
- 所有相关索引被删除
- **DROP TABLE** 语句不能回滚，但是可以闪回

```
DROP TABLE dept80;
Table dropped.
```

改变对象的名称

- 执行RENAME语句改变表，视图，序列，或同义词的名称

```
RENAME dept TO detail_dept;
Table renamed.
```

- 必须是对象的拥有者

清空表

- **TRUNCATE TABLE 语句:**
  - 删除表中所有的数据
  - 释放表的存储空间

```
TRUNCATE TABLE detail_dept;
Table truncated.
```

- **TRUNCATE语句不能回滚**
- **可以使用 DELETE 语句删除数据**

约束

- 约束是表一级的限制
- 如果存在依赖关系，约束可以防止错误的删除数据
- 约束的类型：
  - NOT NULL
  - UNIQUE
  - PRIMARY KEY
  - FOREIGN KEY
  - CHECK

约束规则

- 用户可以自定义约束，也可以使用Oracle Server的sys_cn格式命名约束
- 约束创建的时机：
  - 创建表的时候，同时创建约束
  - 表结构创建完成后
- 约束可以定义在列一级，或者是表一级
- 通过数据字典查看约束

非空约束

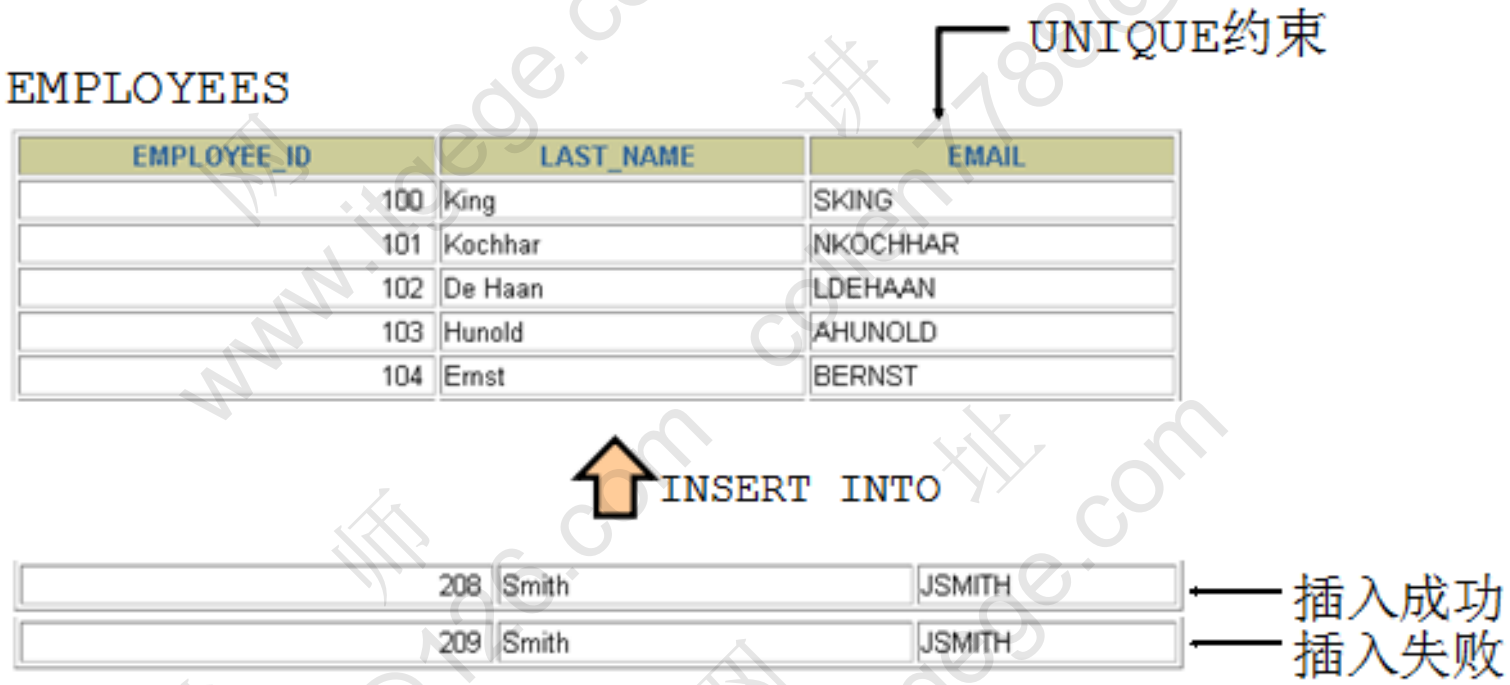
- 保证列的值不能为空

EMPLOYEE_ID	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID
100	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000	90
101	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000	90
102	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000	90
103	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000	60
104	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	6000	60
178	Grant	KGRANT	011.44.1644.429263	24-MAY-99	SA_REP	7000	
200	Whalen	JWHALEN	515.123.4444	17-SEP-87	AD_ASST	4400	10

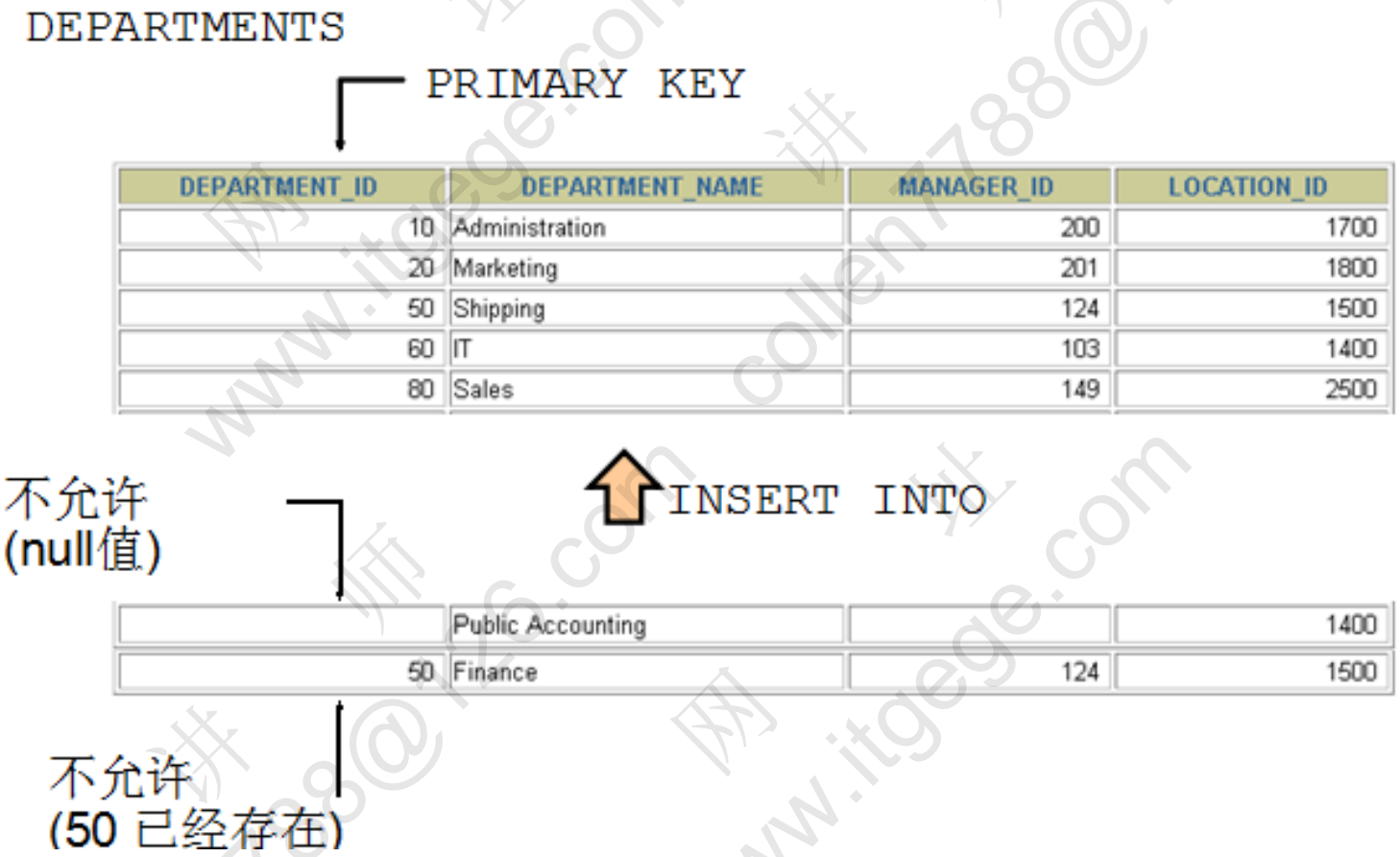
20 rows selected.

NOT NULL约束　　NOT NULL约束　　没有定义NOT NULL 约束

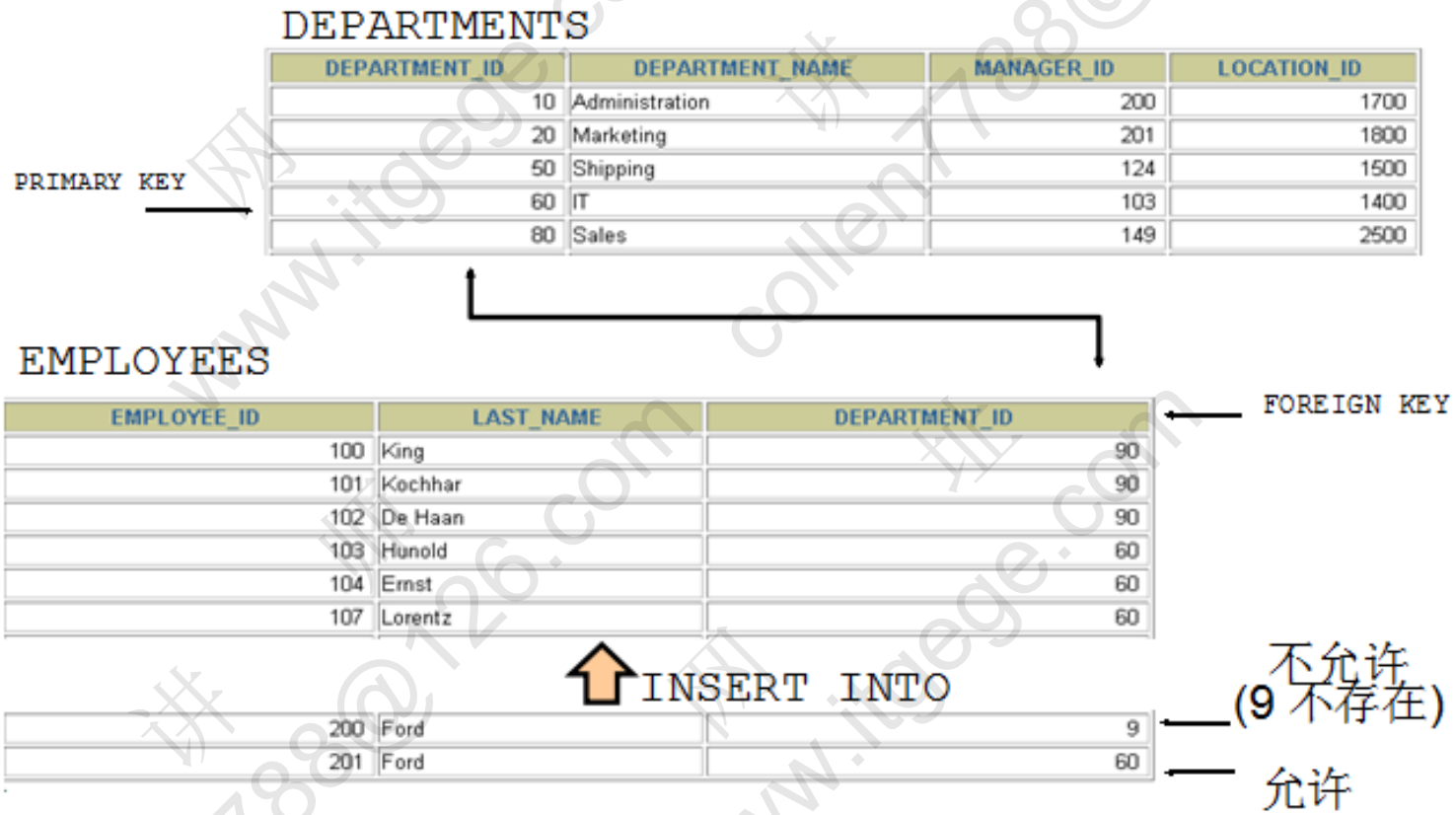
唯一性约束



主键约束



外键约束


DEPARTMENTS
DEPARTMENT_ID
DEPARTMENT_NAME
MANAGER_ID
LOCATION_ID
10 Administration 200 1700
20 Marketing 201 1800
50 Shipping 124 1500
60 IT 103 1400
80 Sales 149 2500
PRIMARY KEY
EMPLOYEES
EMPLOYEE_ID
LAST_NAME
DEPARTMENT_ID
FOREIGN KEY
100 King 90
101 Kochhar 90
102 De Haan 90
103 Hunold 60
104 Ernst 60
107 Lorentz 60
INSERT INTO
200 Ford 9
不允许
(9 不存在)
201 Ford 60
允许

外键约束（续）

- **FOREIGN KEY:** 在子表中，定义了一个表级的约束
- **REFERENCES:** 指定表和父表中的列
- **ON DELETE CASCADE:** 当删除父表时，级联删除子表记录
- **ON DELETE SET NULL:** 将子表的相关依赖记录的外键值置为**null**

check约束

- 定义每一行记录所必须满足的条件
- 下面的表达式可以使用在**check**约束中：
  - 引用CURRVAL, NEXTVAL, LEVEL, 和ROWNUM
  - 调用SYSDATE, UID, USER, 和USERENV 函数
  - 另一个表的查询记录

```
..., salary NUMBER(2)
     CONSTRAINT emp_salary_min
          CHECK (salary > 0),...
```

示例：创建表

```
CREATE TABLE employees
    ( employee_id    NUMBER(6)
      CONSTRAINT     emp_employee_id   PRIMARY KEY
    , first_name     VARCHAR2(20)
    , last_name      VARCHAR2(25)
      CONSTRAINT     emp_last_name_nn  NOT NULL
    , email          VARCHAR2(25)
      CONSTRAINT     emp_email_nn      NOT NULL
      CONSTRAINT     emp_email_uk      UNIQUE
    , phone_number   VARCHAR2(20)
    , hire_date      DATE
      CONSTRAINT     emp_hire_date_nn  NOT NULL
    , job_id         VARCHAR2(10)
      CONSTRAINT     emp_job_nn        NOT NULL
    , salary         NUMBER(8,2)
      CONSTRAINT     emp_salary_ck     CHECK (salary>0)
    , commission_pct NUMBER(2,2)
    , manager_id     NUMBER(6)
    , department_id  NUMBER(4)
      CONSTRAINT     emp_dept_fk       REFERENCES
        departments (department_id));
```

总结

- 通过本章学习，您已经学会如何使用DDL语句创建，修改，删除，和重命名表.

语句	描述
CREATE TABLE	创建表
ALTER TABLE	修改表结构
DROP TABLE	删除表
RENAME	重命名表
TRUNCATE	删除表中的所有数据，并释放存储空间

- 约束

Thank you