幻灯片 1


14
SQL 练习
中国科学院西安网络中心 编译 2005
ORACLE
Copyright © Oracle Corporation, 2001. All rights reserved.

幻灯片 2

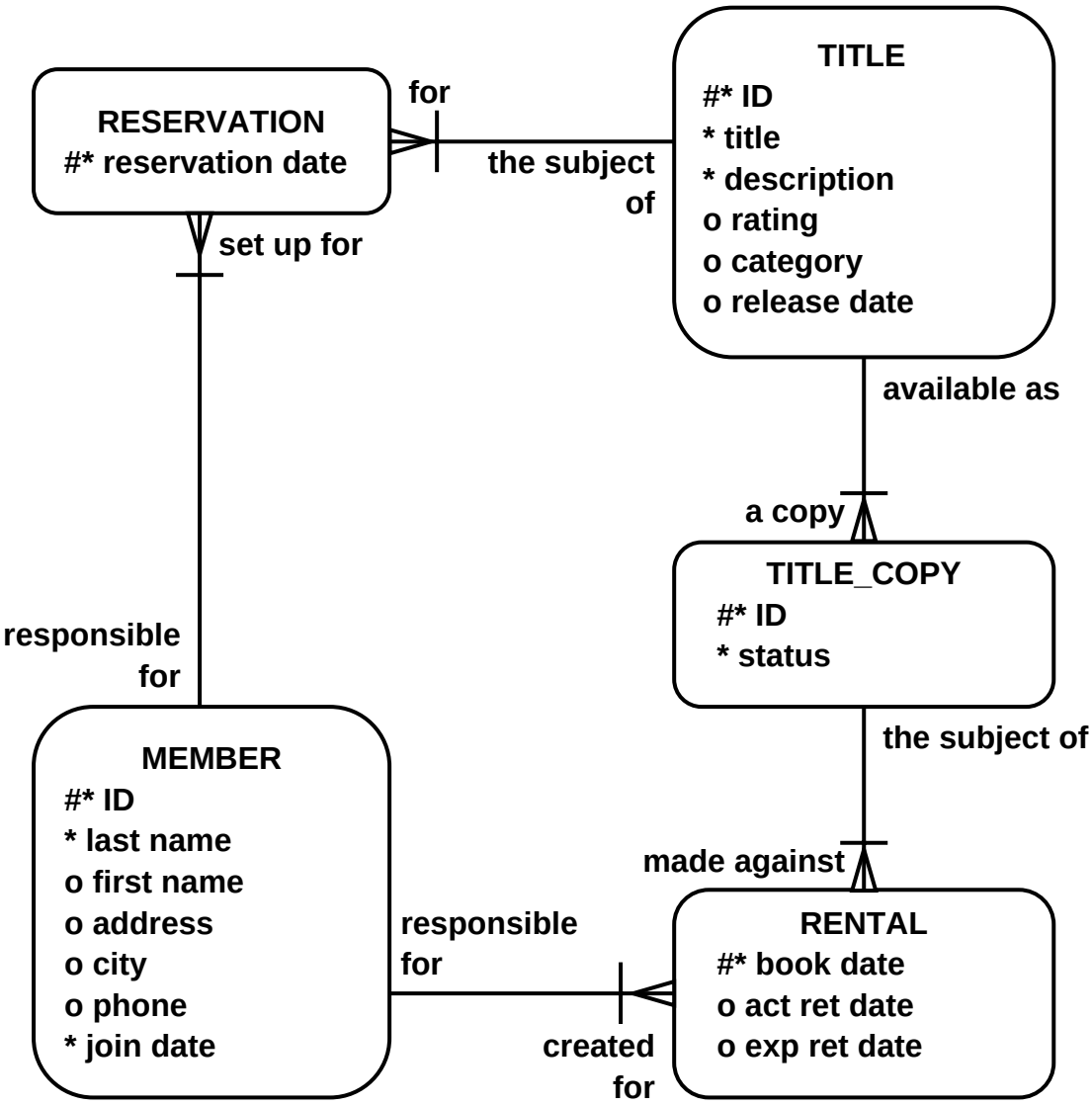
练习概览

在这部分练习中，你将构建一组用于视频应用程序的数据库表，在你创建这些表后，你将插入、更新和删除在一个视频存储数据库中的记录，并生成一个报表，数据库只包含了基本的表。

注：如果你想要构建这些表，你可以在 *i*SQL*Plus 中执行 buildtab.sql 脚本中的命令。如果你想要删除这些表，你可以在 *i*SQL*Plus 中执行 dropvid.sql 脚本中的命令。然后，你可以在 *i*SQL*Plus 中执行 buildvid.sql 脚本中的命令来创建并构成表。如果你使用 buildvid.sql 脚本创建和构成表，从步骤 6b 开始。

幻灯片 3

视频应用程序的实体关系图



练习 14

1. 基于下面的举例图表创建表，选择适当的数据类型并确认添加完整性约束。

a. 表名：MEMBER

Column_Name	MEMBER_ID	LAST_NAME	FIRST_NAME	ADDRESS	CITY	PHONE	JOIN_DATE
Key Type	PK						
Null/ Unique	NN,U	NN					NN
Default Value							System Date
Data Type	NUMBER	VARCHAR2	VARCHAR2	VARCHAR2	VARCHAR2	VARCHAR2	DATE
Length	10	25	25	100	30	15	

```
CREATE TABLE member
(member_id          NUMBER(10)
                    CONSTRAINT member_member_id_pk PRIMARY KEY,
last_name          VARCHAR2(25)
                    CONSTRAINT member_last_name_nn NOT NULL,
first_name         VARCHAR2(25),
address            VARCHAR2(100),
city               VARCHAR2(30),
phone              VARCHAR2(15),
join_date          DATE DEFAULT SYSDATE
                    CONSTRAINT member_join_date_nn NOT NULL);
```

b. 表名：TITLE

Column_Name	TITLE_ID	TITLE	DESCRIPTION	RATING	CATEGORY	RELEASE_DATE
Key Type	PK					
Null/ Unique	NN,U	NN	NN			
Check				G, PG, R, NC17, NR	DRAMA, COMEDY, ACTION, CHILD, SCIFI, DOCUMENTARY	
Data Type	NUMBER	VARCHAR2	VARCHAR2	VARCHAR2	VARCHAR2	DATE
Length	10	60	400	4	20	

```
CREATE TABLE title
(title_id       NUMBER(10)
                CONSTRAINT title_title_id_pk PRIMARY KEY,
title           VARCHAR2(60)
                CONSTRAINT title_title_nn NOT NULL,
description     VARCHAR2(400)
                CONSTRAINT title_description_nn NOT NULL,
rating          VARCHAR2(4)
                CONSTRAINT title_rating_ck CHECK
                (rating IN ('G', 'PG', 'R', 'NC17', 'NR')),
category        VARCHAR2(20),
                CONSTRAINT title_category_ck CHECK
                (category IN ('DRAMA', 'COMEDY', 'ACTION',
                         'CHILD', 'SCIFI', 'DOCUMENTARY')),
release_date    DATE);
```

c. 表名：TITLE_COPY

Column_Name	COPY_ID	TITLE_ID	STATUS
Key Type	PK	PK,FK	
Null/Unique	NN,U	NN,U	NN
Check			AVAILABLE, DESTROYED, RENTED, RESERVED
Data Type	NUMBER	VARCHAR2	VARCHAR2
FK Ref Table		TITLE	
FK Ref Column		TITLE_ID	
Length	10	10	15

```
CREATE TABLE title_copy
(copy_id     NUMBER(10),
title_id     NUMBER(10)
   CONSTRAINT title_copy_title_if_fk REFERENCES title(title_id),
status       VARCHAR2(15)
   CONSTRAINT title_copy_status_nn NOT NULL CONSTRAINT
   title_copy_status_ck CHECK
   (status IN ('AVAILABLE', 'DESTROYED','RENTED', 'RESERVED')),
   CONSTRAINT title_copy_copy_id_title_id_pk
                              PRIMARY KEY (copy_id, title_id));
```

d. 表名：RENTAL

Column_ Name	BOOK_ DATE	MEMBER_ ID	COPY_ ID	ACT_RET_ DATE	EXP_RET_ DATE	TITLE_ ID
Key Type	PK	PK,FK1	PK,FK2			PK,FK2
Null/ Unique	NN,U	NN	NN			
Default Value	System Date			System Date + 2 days		
FK Ref Table		MEMBER	TITLE_ COPY			TITLE_ COPY
FK Ref Column		MEMBER_ ID	COPY_ ID			TITLE_ID
Data Type	DATE	NUMBER	NUMBER	DATE	DATE	NUMBER
Length		10	10			10

```
CREATE TABLE rental
(book_date      DATE DEFAULT SYSDATE,
member_id       NUMBER(10) CONSTRAINT rental_member_id_fk
REFERENCES      member(member_id),
copy_id         NUMBER(10),
act_ret_date    DATE,
exp_ret_date    DATE DEFAULT SYSDATE + 2,
title_id        NUMBER(10),
                CONSTRAINT rental_book_date_copy_title_pk
                PRIMARY KEY (book_date, member_id,
                copy_id,title_id),
                CONSTRAINT rental_copy_id_title_id_fk FOREIGN KEY
                (copy_id, title_id) REFERENCES title_copy(copy_id,
                title_id));
```

e. 表名：RESERVATION

Column Name	RES_DATE	MEMBER_ID	TITLE_ID
Key Type	PK	PK,FK1	PK,FK2
Null/Unique	NN,U	NN,U	NN
FK Ref Table		MEMBER	TITLE
FK Ref Column		MEMBER_ID	TITLE_ID
Data Type	DATE	NUMBER	NUMBER
Length		10	10

```
CREATE TABLE reservation
(res_date       DATE,
member_id       NUMBER(10)
```

```
                    CONSTRAINT reservation_member_id REFERENCES
                            member(member_id),
    title_id        NUMBER(10)
                    CONSTRAINT reservation_title_id REFERENCES
                    title(title_id),
                    CONSTRAINT reservation_resdate_mem_tit_pk
                    PRIMARY KEY (res_date, member_id, title_id));
```

2. 检查数据字典验证表和约束已被完全创建。

```
SELECT table_name
FROM user_tables
WHERE table_name IN ('MEMBER', 'TITLE', 'TITLE_COPY',
                     'RENTAL', 'RESERVATION');

SELECT constraint_name, constraint_type, table_name
FROM user_constraints
WHERE table_name IN ('MEMBER', 'TITLE', 'TITLE_COPY',
                     'RENTAL', 'RESERVATION');
```

3. 创建序列来唯一地标识在 MEMBER 表和 TITLE 表中的每一个行。

a. MEMBER 表的成员号：从 101 开始；不允许序列值缓存。序列名为 MEMBER_ID_SEQ。

```
CREATE SEQUENCE member_id_seq
START WITH 101
NOCACHE;
```

b. Title number for the TITLE 表的标题号：从 92 开始；不缓存。序列名为：TITLE_ID_SEQ。

```
CREATE SEQUENCE title_id_seq
START WITH 92
NOCACHE;
```

c. 验证在数据字典中已经存在的序列。

```
SELECT sequence_name, increment_by, last_number
FROM user_sequences
WHERE sequence_name IN ('MEMBER_ID_SEQ', 'TITLE_ID_SEQ ');
```

4. 添加数据到表中。为每个要添加的数据集创建一个脚本。

a. 添加电影的标题到 TITLE 表中。写一个脚本输入电影信息。

保存语句在一个命名为 lab14_4a.sql 的脚本中。用序列来唯一地标识每一个标

题。以 DD-MON-YYYY 格式输入发行日期，记住，在字符字段的值必须放在单引号标记中。验证你的添加。

```
SET ECHO OFF
INSERT INTO title(title_id, title, description, rating,
                  category, release_date)
VALUES (title_id_seq.NEXTVAL, 'Willie and Christmas Too',
            'All of Willie''s friends make a Christmas list for
            Santa, but Willie has yet to add his own wish list.',
            'G', 'CHILD', TO_DATE('05-OCT-1995','DD-MON-YYYY')

INSERT INTO title(title_id , title, description, rating,
                  category, release_date)
VALUES (title_id_seq.NEXTVAL, 'Alien Again', 'Yet another
            installment of science fiction history. Can the
            heroine save the planet from the alien life form?',
            'R', 'SCIFI', TO_DATE( '19-MAY-1995','DD-MON-YYYY'))

INSERT INTO title(title_id, title, description, rating,
                  category, release_date)
VALUES (title_id_seq.NEXTVAL, 'The Glob', 'A meteor crashes
            near a small American town and unleashes carnivorous
            goo in this classic.', 'NR', 'SCIFI',
            TO_DATE( '12-AUG-1995','DD-MON-YYYY'))

INSERT INTO title(title_id, title, description, rating,
                  category, release_date)
VALUES (title_id_seq.NEXTVAL, 'My Day Off', 'With a little
            luck and a lot ingenuity, a teenager skips school for
            a day in New York.', 'PG', 'COMEDY',
            TO_DATE( '12-JUL-1995','DD-MON-YYYY'))
...

COMMIT

SET ECHO ON
SELECT title
FROM title;
```

Title	Description	Rating	Category	Release_date
Willie and Christmas Too	All of Willie's friends make a Christmas list for Santa, but Willie has yet to add his own wish list.	G	CHILD	05-OCT-1995

Alien Again	Yet another installation of science fiction history. Can the heroine save the planet from the alien life form?	R	SCIFI	19-MAY-1995
The Glob	A meteor crashes near a small American town and unleashes carnivorous goo in this classic.	NR	SCIFI	12-AUG-1995
My Day Off	With a little luck and a lot of ingenuity, a teenager skips school for a day in New York	PG	COMEDY	12-JUL-1995
Miracles on Ice	A six-year-old has doubts about Santa Claus, but she discovers that miracles really do exist.	PG	DRAMA	12-SEP-1995
Soda Gang	After discovering a cache of drugs, a young couple find themselves pitted against a vicious gang.	NR	ACTION	01-JUN-1995

b. 添加数据到 MEMBER 表中。请将插入语句放在脚本中，并命名为 lab14_4b.sql。执行脚本中的命令。确保使用序列添加成员号。

First_Name	Last_Name	Address	City	Phone	Join_Date
Carmen	Velasquez	283 King Street	Seattle	206-899-6666	08-MAR-1990
LaDoris	Ngao	5 Modrany	Bratislava	586-355-8882	08-MAR-1990
Midori	Nagayama	68 Via Centrale	Sao Paolo	254-852-5764	17-JUN-1991
Mark	Quick-to-See	6921 King Way	Lagos	63-559-7777	07-APR-1990
Audry	Ropeburn	86 Chu Street	Hong Kong	41-559-87	18-JAN-1991
Molly	Urguhart	3035 Laurier	Quebec	418-542-9988	18-JAN-1991

```
SET ECHO OFF
SET VERIFY OFF
INSERT INTO member(member_id, first_name, last_name, address,
                   city, phone, join_date)
VALUES (member_id_seq.NEXTVAL, '&first_name', '&last_name',
            '&address', '&city', '&phone', TO_DATE('&join_date',
            'DD-MM-YYYY');
COMMIT;
SET VERIFY ON
SET ECHO ON
```

c. 添加下面的电影拷贝到 TITLE_COPY 表中：
注：对于该练习有可用的 TITLE_ID 号。

Title	Copy_Id	Status
Willie and Christmas Too	1	AVAILABLE
Alien Again	1	AVAILABLE
	2	RENTED
The Glob	1	AVAILABLE
My Day Off	1	AVAILABLE
	2	AVAILABLE
	3	RENTED
Miracles on Ice	1	AVAILABLE
Soda Gang	1	AVAILABLE

```
INSERT INTO title_copy(copy_id, title_id, status)
VALUES (1, 92, 'AVAILABLE');
INSERT INTO title_copy(copy_id, title_id, status)
VALUES (1, 93, 'AVAILABLE');
INSERT INTO title_copy(copy_id, title_id, status)
VALUES (2, 93, 'RENTED');
INSERT INTO title_copy(copy_id, title_id, status)
VALUES (1, 94, 'AVAILABLE');
INSERT INTO title_copy(copy_id, title_id, status)
VALUES (1, 95, 'AVAILABLE');
INSERT INTO title_copy(copy_id, title_id,status)
VALUES (2, 95, 'AVAILABLE');
INSERT INTO title_copy(copy_id, title_id,status)
VALUES (3, 95, 'RENTED');
INSERT INTO title_copy(copy_id, title_id,status)
VALUES (1, 96, 'AVAILABLE');
INSERT INTO title_copy(copy_id, title_id,status)
VALUES (1, 97, 'AVAILABLE');
```

d. 添加下面的租用到 RENTAL 表中：

注：标题号依赖于序列号，它们可能是不同的。

Title_Id	Copy_Id	Member_Id	Book_date	Exp_Ret_Date	Act_Ret_Date
92	1	101	3 days ago	1 day ago	2 days ago
93	2	101	1 day ago	1 day from now	
95	3	102	2 days ago	Today	
97	1	106	4 days ago	2 days ago	2 days ago

```
INSERT INTO rental(title_id, copy_id, member_id, book_date,
                   exp_ret_date, act_ret_date)
VALUES (92, 1, 101, sysdate-3, sysdate-1, sysdate-2);
INSERT INTO rental(title_id, copy_id, member_id, book_date,
                   exp_ret_date, act_ret_date)
```

```
VALUES (93, 2, 101, sysdate-1, sysdate-1, NULL);
INSERT INTO rental(title_id, copy_id, member_id, book_date,
                   exp_ret_date, act_ret_date)
VALUES (95, 3, 102, sysdate-2, sysdate, NULL);
INSERT INTO rental(title_id, copy_id, member_id, book_date,
                   exp_ret_date,act_ret_date)
VALUES (97, 1, 106, sysdate-4, sysdate-2, sysdate-2);
COMMIT;
```

5. 创建一个名为 TITLE_AVAIL 的视图，显示电影标题和每个拷贝的可用性，以及，如果已租出它的预期归还日期。从视图中查询所有行，用标题排序。

注：你的结果可能不同。

```
CREATE VIEW title_avail AS
SELECT    t.title, c.copy_id, c.status, r.exp_ret_date
FROM      title t, title_copy c, rental r
WHERE     t.title_id = c.title_id
AND       c.copy_id = r.copy_id(+)
AND       c.title_id = r.title_id(+);

SELECT    *
FROM      title_avail
ORDER BY  title, copy_id;
```

6. 改变表中的数据。

a. 添加一个新的标题 (**title**) 记录，电影是 “Interstellar Wars” ，该电影的等级为 PG，并且分类为一部科学幻想电影，发行日期是 07-JUL-77。简介为 “Futuristic interstellar action movie. Can the rebels save the humans from the evil empire?”，为两个拷贝分别添加一个标题拷贝表 (**title_copy**) 记录。

```
INSERT INTO title(title_id, title, description, rating, category,
                  release_date)
VALUES (title_id_seq.NEXTVAL, 'Interstellar Wars', 'Futuristic
        interstellar action movie. Can the rebels save the humans
        from the evil Empire?', 'PG', 'SCIFI', '07-JUL-77');
INSERT INTO title_copy (copy_id, title_id, status)
VALUES (1, 98, 'AVAILABLE');
INSERT INTO title_copy (copy_id, title_id, status)
VALUES (2, 98, 'AVAILABLE');
```

b. 输入两个预定记录，一个预定是对 Carmen Velasquez 的，她想要租 “Interstellar Wars”。另一个预定是给 Mark Quick-to-See 的，他想要租 “Soda Gang”。

```
INSERT INTO reservation (res_date, member_id, title_id)
```

```
VALUES (SYSDATE, 101, 98);
INSERT INTO reservation (res_date, member_id, title_id)
VALUES (SYSDATE, 104, 97);
```

c. 客户 Carmen Velasquez 租了电影 “Interstellar Wars” 拷贝 1，从该电影的预定中删除她，在出租表中记录有关信息，允许使用预计归还日期的默认值，用你创建的视图验证出租一被记录在表中。

注：你的结果可能不同。

```
INSERT INTO rental(title_id, copy_id, member_id)
VALUES    (98,1,101);
UPDATE    title_copy
SET       status= 'RENTED'
WHERE     title_id = 98
AND       copy_id = 1;
DELETE
FROM      reservation
WHERE     member_id = 101;
SELECT    *
FROM      title_avail
ORDER BY  title, copy_id;
```

7. 修改表结构。

a. 添加一个 PRICE 列到 TITLE 表中，记录视频的购买价格。该列应该有 8 位数字长度和两位小数。验证你的修改。

```
ALTER TABLE title
ADD (price NUMBER(8,2));
DESCRIBE title
```

b. 创建一个名为 lab14_7b.sql 的脚本，按照下面的列表，写更新语句，更新包含价格的每个视频的记录。运行脚本中的命令。

注：对于本练习有可用的 TITLE_ID 号。

Title	Price
Willie and Christmas Too	25
Alien Again	35
The Glob	35
My Day Off	35
Miracles on Ice	30
Soda Gang	35
Interstellar Wars	29

```
SET ECHO OFF
SET VERIFY OFF
```

```
DEFINE price=
DEFINE title_id=
UPDATE title
SET price = &price
WHERE title_id = &title_id;
SET VERIFY OFF
SET ECHO OFF
```

c. 确认在以后所有 titles 表中的记录都包含价格值，验证该约束。

```
ALTER TABLE title
MODIFY (price CONSTRAINT title_price_nn NOT NULL);
SELECT constraint_name, constraint_type, search_condition
FROM user_constraints
WHERE table_name = 'TITLE';
```

8. 创建严格报告，标题为 Customer History Report，该报告包含每一个客户租视频的历史。报表包括名字、租用的视频、租用的日期，在一个报告期中所有客户租用的总数。保存产生报表的命令到名为 lab14_8.sql 的角本中。

注：你的结果可能不同。

```
SET ECHO OFF
SET VERIFY OFF
TTITLE 'Customer History Report'
BREAK ON member SKIP 1 ON REPORT
SELECT    m.first_name||' '||m.last_name MEMBER, t.title,
          r.book_date, r.act_ret_date - r.book_date DURATION
FROM      member m, title t, rental r
WHERE     r.member_id = m.member_id
AND       r.title_id = t.title_id
ORDER BY  member;

CLEAR BREAK
TTITLE OFF
SET VERIFY ON
SET ECHO ON
```