**CS2102 Database Systems**

**AY 2013/2014 Semester I**

**SQL Lab Handout**

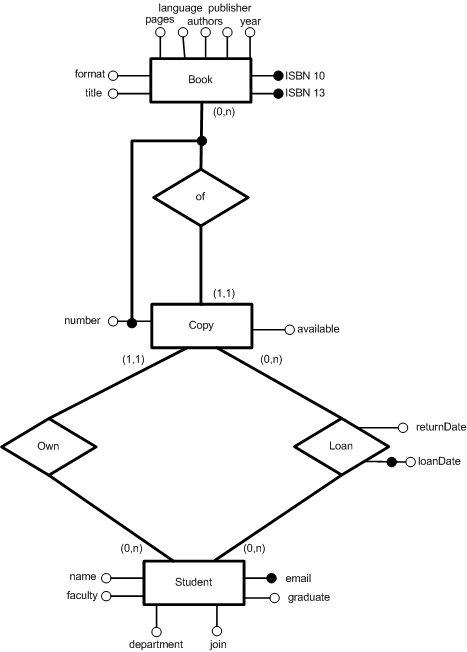
The aim of the lab sessions is to familiarize students with Oracle, syntax of SQL commands and common error messages. Each student will be given an Oracle account. You would have received your Oracle userid and password by now. You can find information about using Oracle at

<http://www.comp.nus.edu.sg/~oradoc>

You can find some notes on the differences between Oracle SQL and standard SQL at

<http://infolab.stanford.edu/~ullman/fcdb/oracle/or-nonstandard.html>

Below is the ER diagram for the Online Book Exchange application (Application 2 in Tutorial 1).



1. Translate the Entity-Relationship diagram of the online book exchange application to a relational schema. Give the SQL DDL statements to create this schema.

CREATE TABLE book (

title VARCHAR(256) NOT NULL,

format CHAR(9) CHECK(format = 'paperback' OR format='hardcover'),

pages INT,

language VARCHAR(32),

authors VARCHAR(256),

publisher VARCHAR(64),

year DATE,

ISBN10 CHAR(10) NOT NULL UNIQUE,

ISBN13 CHAR(14) PRIMARY KEY

);

CREATE TABLE student (

name VARCHAR(32) NOT NULL,

email VARCHAR(256) PRIMARY KEY,

join DATE NOT NULL,

faculty VARCHAR(62) NOT NULL,

department VARCHAR(32) NOT NULL,

graduate DATE,

CHECK(graduate >= join)

);

CREATE TABLE copy (

owner VARCHAR(256) REFERENCES student(email) ON DELETE CASCADE,

ISBN CHAR(14) REFERENCES book(ISBN13) ON DELETE CASCADE,

copynum INT CHECK(copynum>0),

available VARCHAR(6) CHECK(available = 'TRUE' OR available='FALSE'),

PRIMARY KEY (ISBN, copynum)

);

CREATE TABLE loan (

borrower VARCHAR(256) REFERENCES student(email),

ISBN CHAR(14),

copynum INT,

loanDate DATE,

returnDate DATE,

FOREIGN KEY (ISBN, copynum) REFERENCES copy(ISBN, copynum) ON DELETE CASCADE,

PRIMARY KEY (borrower, ISBN, copynum, loanDate),

CHECK(returnDate >= loanDate)

);

1. SQL Warm-up
   1. Create the table Book that contains the following information about books: title, format (paperback or hardcover), number of pages, authors, publisher, year, edition, ISBN-10 and -13. Choose a primary key.

CREATE TABLE book (

title VARCHAR(128) NOT NULL,

format CHAR(9) CHECK(format=’paperback’ OR format=’hardcover’),

pages INT,

authors VARCHAR(128),

publisher VARCHAR(32),

year DATE,

edition INT,

ISBN10 CHAR(10) NOT NULL UNIQUE,

ISBN13 CHAR(14) PRIMARY KEY

);

* 1. Insert a book called “Introduction to Database Systems”. Go to the Web to find the actual details.

INSERT INTO book VALUES ('Introduction to Database Systems', 'paperback', 168, 'Stephane Bressan and Barbara Catania', 'MacGraw-Hill', '2005-01-01', 1, '0071246509', '978-0071246507');

* 1. Insert 3 books authored by C.J. Date. Go to the Web, for instance [www.amazon.com](http://www.amazon.com), to find the details.

INSERT INTO book VALUES ('An Introduction to Database Systems', 'paperback', 1024, 'C.J. Date', 'Addison-Wesley', ‘2003-08-01', 1, '0321197844','978-0321197849');

INSERT INTO book VALUES ('SQL and Relational Theory: How to Write Accurate SQL Code', 'paperback', 432, 'C.J. Date', 'O Reilly Media', '2009-01-23’, 1, '0596523068', '978-0596523060');

* 1. Modify all the books authored by C.J. Date to mention his first name. You can find the author’s first name from the Web.

UPDATE book SET authors=‘Christopher J. Date’ WHERE authors= ‘C. J. Date’;

* 1. Print all the information about books.

SELECT \* FROM book;

* 1. Add a new attribute Language to all the books and set the default language to English.

ALTER TABLE book

ADD language VARCHAR(32) DEFAULT ‘English’;

* 1. Delete all the books authored by C.J. Date.

DELETE FROM book WHERE authors=’Christopher J. Date’;

DELETE FROM book WHERE authors LIKE ‘C%Date’;

* 1. Delete the Book table.

DROP TABLE book;

1. Create the tables for the remainder of this lab using the code in NUSStASchema.sql. You may need to rearrange the statements in the file if you encounter errors.

Populate the tables using NUSStAData.sql.

Check the foreign key constraints to decide in which order to insert the data.

You can clean up data and tables anytime using the file NUSStAClean.sql.

1. Basic SQL queries
   1. Find the names and emails of students.

SELECT name, email FROM student;

* 1. Print the names of students in descending alphabetical order.

SELECT name FROM student

ORDER BY name DESC;

* 1. Are there students with the same name?

SELECT \* FROM student s1, student s2 WHERE s1.name=s2.name AND s1.email < s2.email;

* 1. Find the names and emails of students who have borrowed some books on Database.

SELECT s.name, s.email FROM student s, loan l WHERE l.borrower=s.email;

* 1. Find the names of students who owned a copy of book ‘Pride And Prejudice’.

SELECT s.name FROM student s, copy c, book b WHERE c.owner=s.email AND c.ISBN=b.ISBN13 AND b.title=’ Pride And Prejudice’;

* 1. Find the different names of students who owned a copy of a book other than of ‘978-0262033848’.

SELECT DISTINCT s.name

FROM student s, copy c

WHERE c.owner=s.email AND c.ISBN <> ‘978-0262033848’;

* 1. Find the number of A4 pages needed to photocopy the two books with ISBN-13 '978-0262033848' and '978-0321295354' (2 pages of a book can be copied on one A4 page).

SELECT (b1.pages + b2.pages)/2FROM book b1, book b2WHERE b1.ISBN13 = '978-0262033848' AND b2.ISBN13='978-0321295354';

* 1. Find the names of the students who have borrowed some book by Jane Austen.

SELECT s.name

FROM student s, loan l, book b

WHERE l.borrower=s.email AND l.ISBN=b.ISBN13 AND b.authors='Jane Austen';

1. Queries involving set and aggregate operators, nested queries
   1. Find the names of students who owned or borrowed a copy of book ‘978-0262033848’. Use UNION.

SELECT s.name

FROM student s, copy c

WHERE c.owner=s.email AND c.ISBN=‘978-0262033848’

UNION

SELECT s.name

FROM student s, loan l

WHERE l.borrower=s.email AND l.ISBN=‘978-0262033848’;

* 1. Find the names of students who owned or borrowed a copy of book ‘978-0262033848’. USE OR.

SELECT s.name

FROM student s, copy c, loan l

WHERE (c.owner=s.email AND c.ISBN=‘978-0262033848’)

OR (l.borrower=s.email AND l.ISBN=‘978-0262033848’);

* 1. Delete all the data in table Loan.

DELETE \* FROM loan;

* 1. Try again the last two queries.

We see that the query with OR returns no results. This is wrong. That means that the query was not correct. It happens because the Cartesian product in the FROM clause is empty. When is it correct to use OR? OR should be used among conditions on the exactly same tables.

* 1. Find the total number of copies.

SELECT COUNT(\*)

FROM copy c;

* 1. Find, for each book, the corresponding number of copies. [Print the primary key of the book and the number of copies.]

SELECT c.ISBN, COUNT(\*)

FROM copy c

GROUP BY c.ISBN;

* 1. Find the available books with the largest number of copies.

SELECT c.ISBN

FROM copy c

WHERE c.available='TRUE'

GROUP BY c.ISBN

HAVING COUNT(\*) >= ALL ( SELECT COUNT(\*)

FROM copy c

WHERE c.available='TRUE'

GROUP BY c.ISBN);