# Lab 6: *Query in a multiple linked database: Application*

**Save the completed note as “Lab6\_lastname.doc” (e.g., Lab6\_You.doc). Then, upload this to MyClasses**

**1. Sales data**

1.1. Creating Tables

|  |
| --- |
| CREATE TABLE sale (  saleno INTEGER not null,  saledate DATETIME,  saletext VARCHAR(50),  PRIMARY KEY(saleno));  CREATE TABLE item (  itemno INTEGER not null,  itemname VARCHAR(30),  itemtype CHAR(1),  itemcolor VARCHAR(10),  PRIMARY KEY(itemno));  CREATE TABLE lineitem (  linenum INTEGER not null,  lineqty INTEGER,  lineprice DECIMAL(7,2),  saleno INTEGER,  itemno INTEGER,  PRIMARY KEY(linenum,saleno),  **CONSTRAINT fk\_has\_sale** FOREIGN KEY(saleno) REFERENCES sale(saleno),  **CONSTRAINT fk\_has\_item** FOREIGN KEY(itemno) REFERENCES item(itemno)); |

1.2. Add the rows to the tables.

|  |
| --- |
| insert into sale values (1,'2003-01-15','Scruffy Australian - called himself Bruce');  insert into sale values (2,'2003-01-15','Man. Rather fond of hats.');  insert into sale values (3,'2003-01-15','Woman. Planning to row Atlantic - lengthwise!');  insert into sale values (4,'2003-01-15','Man. Trip to New York - thinks NY is a jungle!');  insert into sale values (5,'2003-01-16','Expedition leader for African safari.');  insert into item values (1,'Pocket knife - Nile','E','Brown');  insert into item values (2,'Pocket knife - Avon','E','Brown');  insert into item values (3,'Compass','N','-');  insert into item values (4,'Geo positioning system','N','-');  insert into item values (5,'Map measure','N','-');  insert into item values (6,'Hat - Polar explorer','C','Red');  insert into item values (7,'Hat - Polar explorer','C','White');  insert into item values (8,'Boots - snake proof','C','Green');  insert into item values (9,'Boots - snake proof','C','Black');  insert into item values (10,'Safari chair','F','Khaki');  insert into item values (11,'Hammock','F','Khaki');  insert into item values (12,'Tent - 8 person','F','Khaki');  insert into item values (13,'Tent - 2 person','F','Khaki');  insert into item values (14,'Safari cooking kit','E','-');  insert into item values (15,'Pith helmet','C','Khaki');  insert into item values (16,'Pith helmet','C','White');  insert into item values (17,'Map case','N','Brown');  insert into item values (18,'Sextant','N','-');  insert into item values (19,'Stetson','C','Black');  insert into item values (20,'Stetson','C','Brown');  insert into lineitem values (1,1,4.5,1,2);  insert into lineitem values (1,1,25,2,6);  insert into lineitem values (2,1,20,2,16);  insert into lineitem values (3,1,25,2,19);  insert into lineitem values (4,1,2.25,2,2);  insert into lineitem values (1,1,500,3,4);  insert into lineitem values (2,1,2.25,3,2);  insert into lineitem values (1,1,500,4,4);  insert into lineitem values (2,1,65,4,9);  insert into lineitem values (3,1,60,4,13);  insert into lineitem values (4,1,75,4,14);  insert into lineitem values (5,1,10,4,3);  insert into lineitem values (6,1,2.25,4,2);  insert into lineitem values (1,50,36,5,10);  insert into lineitem values (2,50,40.5,5,11);  insert into lineitem values (3,8,153,5,12);  insert into lineitem values (4,1,60,5,13);  insert into lineitem values (5,1,0,5,2); |

1.3. Practice Queries

(1) What is the difference between

SELECT \* FROM sale, lineitem, item

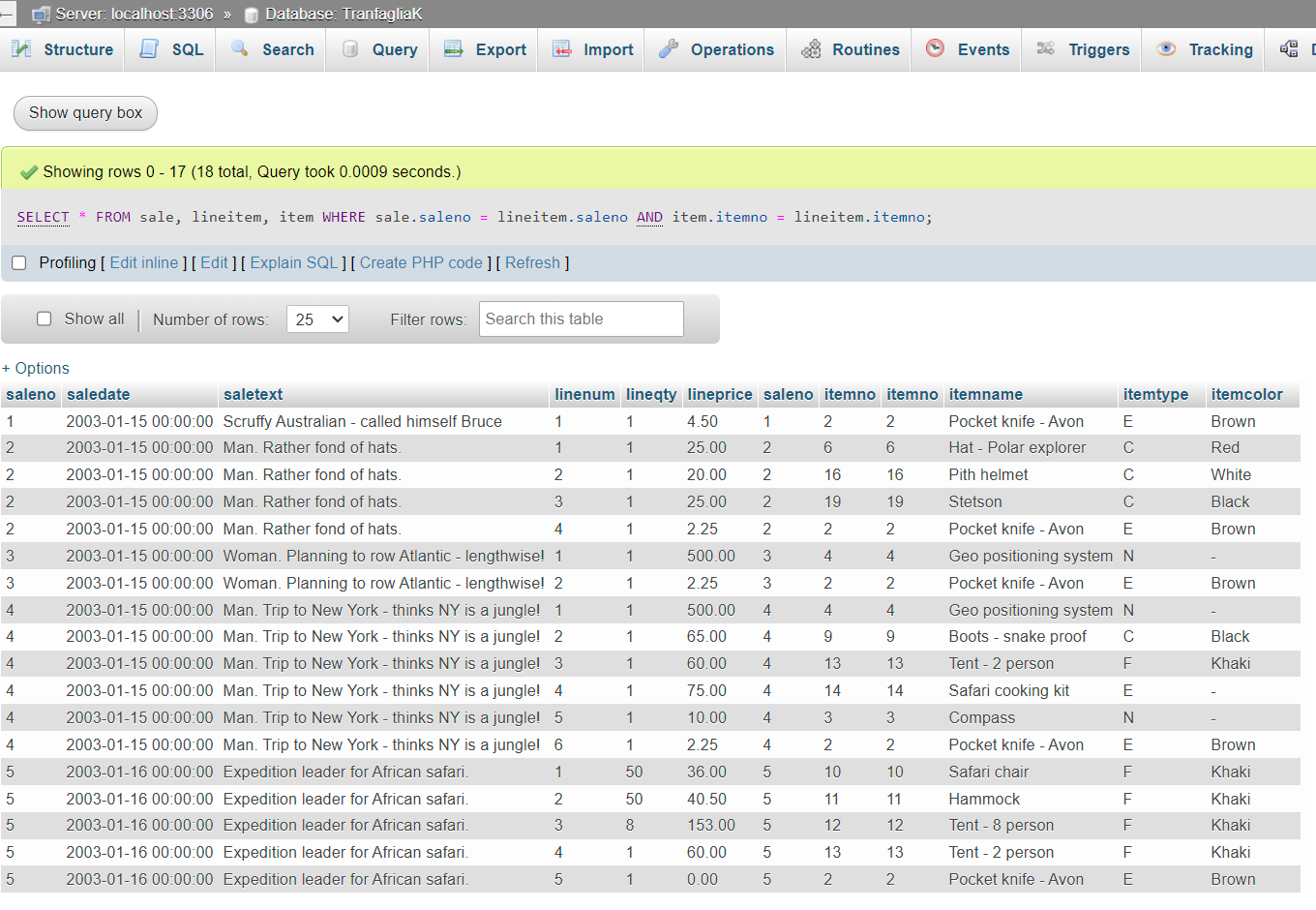
and

SELECT \* FROM sale, lineitem, item

WHERE **sale.saleno = lineitem.saleno**

AND **item.itemno = lineitem.itemno;**

|  |
| --- |
| Cartesian products  Vs.  Joined results  SELECT \* FROM sale, lineitem, item;  This query performs a Cartesian product, or rather, a cross join between the tables. This returns all combinations of the rows from sale, lineitem, and item such that every possible combination is retuned without a set condition to filter the results.   SELECT \* FROM sale, lineitem, item  WHERE **sale.saleno = lineitem.saleno**  AND **item.itemno = lineitem.itemno;**  This query is similar such that it selects all columns from the three tables, however, it specifies a condition for the join. The condition joins the sale and lineitems table based on matching saleno, and the item and lineitem table on the matching itemno. Overall, this condition filters the results of the join. |



(2)

|  |
| --- |
| *List the names of items, quantity, and value of items sold on January 16, 2003* |
| SELECT …… lineqty\*lineprice value |
|  |

(3)

|  |
| --- |
| *Report all clothing items (type “C”) for which a sale is recorded* |
|  |

(4)

|  |
| --- |
| *Report all clothing items (type “C”) that have not been sold* |
|  |

**2. Department-Employee data**



1.1. Creating Tables

|  |
| --- |
| DROP TABLE IF EXISTS EMPLOYEE;  DROP TABLE IF EXISTS DEPARTMENT;  CREATE TABLE DEPARTMENT (  DEPT\_NAME VARCHAR(15) NOT NULL,  DEPT\_FLOOR INT,  DEPT\_PHONE INT,  EMP\_NUMBER INT,  PRIMARY KEY(DEPT\_NAME));  CREATE TABLE EMPLOYEE(  EMP\_NUMBER INT NOT NULL,  EMP\_FNAME VARCHAR(10),  EMP\_SALARY DECIMAL(7,0),  DEPT\_NAME VARCHAR(15),  BOSS\_NUMBER INT,  PRIMARY KEY(EMP\_NUMBER),  CONSTRAINT fk\_belong\_dept FOREIGN KEY(DEPT\_NAME)  REFERENCES DEPARTMENT(DEPT\_NAME),  CONSTRAINT fk\_has\_boss FOREIGN KEY (BOSS\_NUMBER)  REFERENCES EMPLOYEE(EMP\_NUMBER)); |

2.2. Add the rows to the tables.

|  |
| --- |
| INSERT INTO DEPARTMENT VALUES ('Management',5,2001,1);  INSERT INTO DEPARTMENT VALUES ('Marketing',1,2002,2);  INSERT INTO DEPARTMENT VALUES ('Accounting',4,2003,5);  INSERT INTO DEPARTMENT VALUES ('Purchasing',4,2004,7);  INSERT INTO DEPARTMENT VALUES ('Personnel',1,2005,9);  **INSERT INTO EMPLOYEE(EMP\_NUMBER, EMP\_FNAME, EMP\_SALARY, DEPT\_NAME)**  **VALUES (1,'Alice',75000,'Management');**  INSERT INTO EMPLOYEE VALUES (2,'Ned',45000,'Marketing',1);  INSERT INTO EMPLOYEE VALUES (3,'Andrew',25000,'Marketing',2);  INSERT INTO EMPLOYEE VALUES (4,'Clare',22000,'Marketing',2);  INSERT INTO EMPLOYEE VALUES (5,'Todd',38000,'Accounting',1);  INSERT INTO EMPLOYEE VALUES (6,'Nancy',22000,'Accounting',5);  INSERT INTO EMPLOYEE VALUES (7,'Brier',43000,'Purchasing',1);  INSERT INTO EMPLOYEE VALUES (8,'Sarah',56000,'Purchasing',7);  INSERT INTO EMPLOYEE VALUES (9,'Sophie',35000,'Personnel',1); |

2.3. Practice Queries

(1) Find the name of Sophie’s boss

|  |
| --- |
|  |

(2) Find the departments where all the employees earn less than their boss.

|  |
| --- |
|  |

(3) List the departments having an average salary greater than $25,000.

|  |
| --- |
|  |

(4) List the names and manager of the employees of the Marketing department who have a salary greater than $25,000.

|  |
| --- |
|  |