

VICTORIA UNIVERSITY OF WELLINGTON  
*Te Whare Wananga o te Upoko o te Ika a Maui*



## EXAMINATIONS — 2009

END-YEAR

COMP302 / SWEN304  
Database Systems / Database System Engineering

## Appendix

- Do not hand this Appendix in.
- Do not write your answers on this Appendix.

### Contents:

Appendix A. COMPANY Database Schema and a Schema Instance  
Appendix B. SQL Reference  
Appendix C. Experiments with Update Trigger on Employee

## APPENDIX A

### COMPANY Database Schema and a Schema Instance

The relational database schema COMPANY consists of the following relation schemas and referential integrity constraints:

- Employee ({empId, empName, salary},{empId})
- Department ({depId, depName, budget, managerId},{depId})  
 With foreign key [managerId]  $\subseteq$  Employee[empId]
- WorksFor ({empId, depId, jobTitle, percentTime},{empId + depId})  
 With foreign keys [empId]  $\subseteq$  Employee[empId] and  
 [depId]  $\subseteq$  Department[depId]
- Client ({cliId, cliName, address, phone, contactEmpId},{cliId})  
 With foreign key [contactEmpId]  $\subseteq$  Employee[empId]

The current state of the COMPANY database consists of the following relation instances:

Employee		
empId	empName	salary
301	Sylvester	30,000
302	Tweety	45,000
311	Elmer	50,000
320	Bugs	40,000

Department			
depId	depName	budget	managerId
12	Marketing	1,000,000	302
21	Development	1,200,000	311

WorksFor			
empId	depId	jobTitle	percentTime
301	12	Sales engineer	50
301	21	Programmer	50
302	12	Manager	100
311	21	Manager	100
320	12	Ad writer	100

Client				
cliId	cliName	address	phone	contactEmpId
5005	Marvin	5A Home St	04-555-555	320
5007	Speedy	5A Next St	(null)	301
5009	Ann	5A Home St	04-555-555	320

## APPENDIX B

### SQL Reference:

#### CREATE TABLE

```
CREATE TABLE table_name (
  { column_name data_type [ DEFAULT default_expr ] [ column_constraint [, ... ] ]
  | table_constraint } [, ... ]
)
```

where *column\_constraint* is:

```
[ CONSTRAINT constraint_name ]
{ NOT NULL | NULL | UNIQUE | PRIMARY KEY | CHECK (expression) |
  REFERENCES reftable [ ( refcolumn ) ] [ ON DELETE action ] [ ON UPDATE action ] }
```

*table\_constraint* is:

```
[ CONSTRAINT constraint_name ]
{ UNIQUE ( column_name [, ... ] ) |
  PRIMARY KEY ( column_name [, ... ] ) |
  CHECK ( expression ) |
  FOREIGN KEY ( column_name [, ... ] ) REFERENCES reftable [ ( refcolumn [, ... ] ) ]
  [ ON DELETE action ] [ ON UPDATE action ] }
```

and *action* is one of RESTRICT, CASCADE, SET NULL, or SET DEFAULT

#### ALTER TABLE

```
ALTER TABLE [ ONLY ] name [ * ]
  action [, ... ]
```

where *action* is one of:

```
ADD [ COLUMN ] column type [ column_constraint [ ... ] ]
DROP [ COLUMN ] column [ RESTRICT | CASCADE ]
ALTER [ COLUMN ] column TYPE type [ USING expression ]
ALTER [ COLUMN ] column SET DEFAULT expression
ALTER [ COLUMN ] column DROP DEFAULT
ALTER [ COLUMN ] column { SET | DROP } NOT NULL
ALTER [ COLUMN ] column SET STATISTICS integer
ALTER [ COLUMN ] column SET STORAGE { PLAIN | EXTERNAL | EXTENDED | MAIN }
ADD table_constraint
DROP CONSTRAINT constraint_name [ RESTRICT | CASCADE ]
DISABLE TRIGGER [ trigger_name | ALL | USER ]
ENABLE TRIGGER [ trigger_name | ALL | USER ]
CLUSTER ON index_name
SET WITHOUT CLUSTER
SET WITHOUT OIDS
SET ( storage_parameter = value [, ... ] )
RESET ( storage_parameter [, ... ] )
INHERIT parent_table
NO INHERIT parent_table
OWNER TO new_owner
SET TABLESPACE new_tablespace
```

#### INSERT

```
INSERT INTO table [ ( column [, ... ] ) ]
  { DEFAULT VALUES | VALUES ( { expression | DEFAULT } [, ... ] [, ... ] | query ) }
  [ RETURNING * | output_expression [ AS output_name ] [, ... ] ]
```

**UPDATE**

```
UPDATE [ ONLY ] table [ [ AS ] alias ]
  SET { column = { expression | DEFAULT } |
      ( column [, ...] ) = ( { expression | DEFAULT } [, ...] ) } [, ...]
  [ FROM fromlist ]
  [ WHERE condition ]
  [ RETURNING * | output_expression [ AS output_name ] [, ...] ]
```

**SELECT**

```
SELECT [ ALL | DISTINCT ]
  * | expression [ AS output_name ] [, ...]
  [ FROM from_item [, ...] ]
  [ WHERE condition ]
  [ GROUP BY expression [, ...] ]
  [ HAVING condition [, ...] ]
  [ { UNION | INTERSECT | EXCEPT } [ ALL ] select ]
  [ ORDER BY expression [ ASC | DESC | USING operator ] [, ...] ]
  [ FOR UPDATE [ OF tablename [, ...] ] ]
```

where *from\_item* can be:

```
[ ONLY ] table_name [ * ] [ [ AS ] alias [ ( column_alias_list ) ] ] |
( select ) [ AS ] alias [ ( column_alias_list ) ] |
from_item [ NATURAL ] [ join_type ] JOIN from_item [ ON join_condition | USING ( join_column_list ) ]
```

and *join\_type* can be:

```
INNER |
LEFT [ OUTER ] |
RIGHT [ OUTER ] |
FULL [ OUTER ] |
CROSS
```

For INNER (the default) and OUTER join types, exactly one of NATURAL, ON *join\_condition*, or USING (*join\_column\_list*) must appear. For CROSS JOIN, none of these items may appear.

**CREATE VIEW**

```
CREATE VIEW view [ ( column name list ) ] AS SELECT query
```

**Some Data Types**

integer, int, smallint  
 character[*n*], char[*n*], character varying[*n*], varchar[*n*], varchar  
 numeric, numeric[*precision*], numeric[*precision*, *scale*], real, double  
 boolean, date,

**Note:** [ *xxx* ] means *xxx* is optional, { *xxx* | *yyy* } means *xxx* or *yyy*.

## APPENDIX C

### Experiments with UPDATE\_TRIGGER on EMPLOYEE

#### Update a):

```
*****
company=# update employee set salary = 90000 where empid = 302;
```

```
NOTICE:      Due to Referential Integrity Constraint, update
propagated to client table!
```

```
UPDATE 1
```

```
*****
company=# select * from employee;
```

empid	empname	salary
301	Sylvester	30000
320	Bugs	40000
311	Elmer	50000
302	Tweety	90000

```
(4 rows)
```

```
company=# select * from client;
```

cliid	cliname	address	phone	contactempid
5007	Speedy	5A Next St		301
5005	Marvin	5A Home St	04-555-555	320
5009	Ann	5A Home St	04-555-555	320

```
(3 rows)
```

#### Update b):

```
*****
company=# update employee set empid = 555 where empid = 311;
```

```
NOTICE:      Due to Referential Integrity Constraint, update
propagated to client table!
```

```
UPDATE 1
```

```
*****
company=# select * from employee;
```

empid	empname	salary
301	Sylvester	30000
320	Bugs	40000
302	Tweety	90000
555	Elmer	50000

```
(4 rows)
```

```
company=# select * from client;
```

cliid	cliname	address	phone	contactempid
5007	Speedy	5A Next St		301
5005	Marvin	5A Home St	04-555-555	320
5009	Ann	5A Home St	04-555-555	320

```
(3 rows)
```

**Update c):**

```

*****
company=# update employee set empid = 999 where empid = 320;
NOTICE:      Due to Referential Integrity Constraint, update
propagated to client table!
UPDATE 1
*****
company=# select * from employee;
 empid | empname | salary
-----+-----+-----
    301 | Sylvester |   30000
    302 | Tweety   |   90000
    555 | Elmer    |   50000
    999 | Bugs     |   40000
(4 rows)

```

```

company=# select * from client;
 cliid | cliname | address | phone | contactempid
-----+-----+-----+-----+-----
    5007 | Speedy | 5A Next St |      |           301
    5005 | Marvin  | 5A Home St | 04-555-555 |           999
    5009 | Ann   | 5A Home St | 04-555-555 |           999
(3 rows)

```

**Update d):**

```

*****
company=# update employee set empid = 333 where empid = 777;
UPDATE 0
*****
company=# select * from employee;
 empid | empname | salary
-----+-----+-----
    301 | Sylvester |   30000
    302 | Tweety   |   90000
    555 | Elmer    |   50000
    999 | Bugs     |   40000
(4 rows)

```

```

company=# select * from client;
 cliid | cliname | address | phone | contactempid
-----+-----+-----+-----+-----
    5007 | Speedy | 5A Next St |      |           301
    5005 | Marvin  | 5A Home St | 04-555-555 |           999
    5009 | Ann   | 5A Home St | 04-555-555 |           999
(3 rows)

```