INFT3007 – The Information Resource

Assignment 3

Table of Contents

Introduction

The aim of this assignment is to enable students to gain a better understanding of implementing a RDMS (Relational Database Management System):

- The students are given the required test data input for each table, for which they will implement the appropriate tables (with correct data types) into a database system of choice. The students are then given a set of data querying tasks to perform, involving deciphering the business requirements, and devising a solution written in SQL.

- The database system is then required to be given a web-based interface, enabling the ability to execute and display the output for each conjured query. The web interface will give students reasoning for considering using open source approaches, also providing them a better understanding of web-based systems, and how they interact with databases to create dynamic content.

Task 1

Provide basic listings showing that your database has been created. You should include metadata from all tables and a basic data dictionary explaining at least five of the less obvious attributes. All tables should be in 3NF or BCNF.

```
mysql> use currency_loans;
Database changed
mysql> show tables;
+--------------------------+
| Tables_in_currency_loans |
+--------------------------+
| branch                   |
| client                   |
| contract                 |
| currency                 |
| group                    |
| industry                 |
| large_loans              |
| officer                  |
| payment                  |
+--------------------------+
9 rows in set (0.00 sec)

mysql> DESCRIBE branch;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| br_code | char(3)     | NO   | PRI | NULL    |       |
| br_name | varchar(25) | NO   |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

```
mysql> DESCRIBE client;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| cl_code       | varchar(5)  | NO   | PRI | NULL    |       |
| cl_name       | varchar(25) | NO   |     | NULL    |       |
| ic_code       | int(11)     | NO   | MUL | NULL    |       |
| cl_limit      | int(10)     | NO   |     | NULL    |       |
| cl_limit_curr | char(3)     | NO   | MUL | NULL    |       |
| gr_code       | char(4)     | YES  | MUL | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql> DESCRIBE contract;
+-----------+------------+------+-----+---------+-------+
| Field     | Type       | Null | Key | Default | Extra |
+-----------+------------+------+-----+---------+-------+
| con_id    | int(11)    | NO   | PRI | NULL    |       |
| con_date  | date       | NO   |     | NULL    |       |
| cl_code   | varchar(5) | NO   | MUL | NULL    |       |
| off_id    | int(11)    | NO   | MUL | NULL    |       |
| br_code   | varchar(3) | NO   | MUL | NULL    |       |
| con_amnt  | int(10)    | NO   |     | NULL    |       |
| curr_code | char(3)    | NO   | MUL | NULL    |       |
+-----------+------------+------+-----+---------+-------+
7 rows in set (0.00 sec)

mysql> DESCRIBE currency;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| curr_code    | char(3)      | NO   | PRI | NULL    |       |
| curr_name    | varchar(25)  | NO   |     | NULL    |       |
| usd_xchg_rate | decimal(10,6) | NO   |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

```
mysql> DESCRIBE `group`;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| gr_code      | char(4)     | NO   | PRI | NULL    |       |
| gr_name      | varchar(25) | NO   |     | NULL    |       |
| gr_limit     | int(10)     | NO   |     | NULL    |       |
| gr_limit_curr | char(3)    | NO   | MUL | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> DESCRIBE industry;
+---------+-------------+------+-----+---------+----------------+
| Field   | Type        | Null | Key | Default | Extra          |
+---------+-------------+------+-----+---------+----------------+
| ic_code | int(11)     | NO   | PRI | NULL    | auto_increment |
| ic_name | varchar(15) | NO   |     | NULL    |                |
+---------+-------------+------+-----+---------+----------------+
2 rows in set (0.00 sec)

mysql> DESCRIBE officer;
+------------+-------------+------+-----+---------+----------------+
| Field      | Type        | Null | Key | Default | Extra          |
+------------+-------------+------+-----+---------+----------------+
| off_id     | int(11)     | NO   | PRI | NULL    | auto_increment |
| off_name   | varchar(25) | NO   |     | NULL    |                |
| position   | varchar(25) | NO   |     | NULL    |                |
| br_code    | char(3)     | NO   | MUL | NULL    |                |
| supervisor | int(11)     | NO   | MUL | NULL    |                |
+------------+-------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

```
mysql> DESCRIBE payment;
+-------------------+---------+------+-----+---------+----------------+
| Field             | Type    | Null | Key | Default | Extra          |
+-------------------+---------+------+-----+---------+----------------+
| payment_id        | int(11) | NO   | PRI | NULL    | auto_increment |
| con_id            | int(11) | NO   | MUL | NULL    |                |
| payment_due_date  | date    | NO   |     | NULL    |                |
| payment_amnt      | int(10) | NO   |     | NULL    |                |
| payment_made_date | date    | YES  |     | NULL    |                |
+-------------------+---------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

Data Dictionary

Branch

Field	Type	Null	Default	Comments
br_code	char(3)	No		The branch primary identifier.
br_name	varchar(25)	No		The branch name.

Client

Field	Type	Null	Default	Comments
cl_code	varchar(5)	No		The client primary identifier.
cl_name	varchar(25)	No		The company name of the client.
ic_code	int(11)	No		The industry foreign key. The industry being the classification the client company falls under.
cl_limit	int(10)	No		The exposure limit; being the max amount that can be borrowed.
cl_limit_curr	char(3)	No		The currency foreign key, relating to the type of currency used to define the client limit.
gr_code	char(4)	Yes	*NULL*	The group foreign key, enabling the ability to have client companies relating to each other.

Contract

Field	Type	Null	Default	Comments
con_id	int(11)	No		The contract primary identifier.
con_date	date	No		The date the contract was made.
cl_code	varchar(5)	No		Client foreign key. The client who made the contract.
off_id	int(11)	No		Officer foreign key. The lending officer who is responsible for setting up the contract.
br_code	varchar(3)	No		Branch foreign key. The branch where the contract was made/ created.
con_amnt	int(10)	No		The contract amount.
curr_code	char(3)	No		The currency used to define the contract amount.

Currency

Field	Type	Null	Default	Comments
curr_code	char(3)	No		The currency primary identifier.
curr_name	varchar(25)	No		The full currency name.
usd_xchg_rate	decimal(10,6)	No		Exchange rate relative to the USD.

Group

Field	Type	Null	Default	Comments
gr_code	char(4)	No		The group primary identifier.
gr_name	varchar(25)	No		The name of the cooperate group.
gr_limit	int(10)	No		The maximum amount allowed for the group, including all children clients.
gr_limit_curr	char(3)	No		Currency used to define the group limit.

Industry

Field	Type	Null	Default	Comments
ic_code	int(11)	No		The industry primary identifier.
ic_name	varchar(15)	No		Industry's name.

Officer

Field	Type	Null	Default	Comments
off_id	int(11)	No		The officer primary identifier.
off_name	varchar(25)	No		Officer name.
position	varchar(25)	No		Position role.
br_code	char(3)	No		Branch foreign key. The officer home branch.
supervisor	int(11)	No		Officer supervisor (unary relationship). The senior officer which the officer reports to. In the case that the officer doesn't report to a superior it will be set to themselves.

Payment

Field	Type	Null	Default	Comments
payment_id	int(11)	No		The payment primary identifier.
con_id	int(11)	No		Contract foreign key.
payment_due_date	date	No		Payment due date.
payment_amnt	int(10)	No		Payment amount due.
payment_made_date	date	Yes	*NULL*	Payment made date (otherwise null).

Task 2

Provide a listing of the data contents of each table.

```
mysql> SELECT * FROM branch;
+---------+------------+
| br_code | br_name    |
+---------+------------+
| AUC     | Auckland   |
| BRI     | Brisbane   |
| MEL     | Melbourne  |
| SYD     | Sydney     |
| WEL     | Wellington |
+---------+------------+
5 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM client;
+---------+----------------------+---------+----------+---------------+---------+
| cl_code | cl_name              | ic_code | cl_limit | cl_limit_curr | gr_code |
+---------+----------------------+---------+----------+---------------+---------+
| AEI     | Aust Electronics Inc |       5 | 12000000 | USD           | NULL    |
| ARH     | Arrow Holdings Ltd   |       5 | 15000000 | USD           | ARRG    |
| ARI     | Arrow International  |       5 | 10000000 | USD           | ARRG    |
| ARR     | Arrow Ltd            |       5 | 20000000 | USD           | ARRG    |
| BCC     | Barton Chemical Inc  |       4 | 40000000 | USD           | NULL    |
| DIFF    | Differentials Inc    |       2 |  4000000 | USD           | NULL    |
| HAR     | Harrow (NZ) Ltd      |       3 | 12000000 | USD           | HARG    |
| HARA    | Harrow Australasia   |       3 | 15000000 | USD           | HARG    |
| MTK     | Martek Mining Inc    |       1 |  5000000 | USD           | MTKG    |
| MTKE    | Martek Exploration   |       1 |  3000000 | USD           | MTKG    |
| NZT     | New Zealand Tyres    |       2 |  4000000 | USD           | NULL    |
+---------+----------------------+---------+----------+---------------+---------+
11 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM contract;
+--------+------------+---------+--------+---------+----------+-----------+
| con_id | con_date   | cl_code | off_id | br_code | con_amnt | curr_code |
+--------+------------+---------+--------+---------+----------+-----------+
|      5 | 2009-04-12 | MTK     |     10 | SYD     |  3000000 | NZD       |
|     10 | 2009-06-01 | ARI     |      5 | SYD     | 12000000 | NZD       |
|     15 | 2009-12-04 | DIFF    |      8 | MEL     |  3000000 | SGD       |
|     20 | 2010-02-15 | ARH     |      5 | MEL     | 20000000 | AUD       |
|     25 | 2010-02-18 | HAR     |      9 | AUC     |  9000000 | SGD       |
|     30 | 2010-04-12 | ARI     |      6 | SYD     | 11000000 | NZD       |
|     40 | 2010-04-15 | AEI     |      8 | MEL     |  2000000 | GBP       |
|     50 | 2010-04-20 | MTKE    |     10 | AUC     |  4000000 | USD       |
|     60 | 2010-04-28 | AEI     |      6 | SYD     |  3000000 | USD       |
+--------+------------+---------+--------+---------+----------+-----------+
9 rows in set (0.00 sec)


mysql> SELECT * FROM currency;
+-----------+----------------------+---------------+
| curr_code | curr_name            | usd_xchg_rate |
+-----------+----------------------+---------------+
| AUD       | Australian Dollars   |      0.914843 |
| EUR       | Euros                |      1.272300 |
| GBP       | United Kingdom Pounds |      1.538900 |
| JPY       | Japanese Yen         |      0.011966 |
| NOK       | Norwegian Krone      |      0.161349 |
| NZD       | New Zealand Dollars  |      0.719116 |
| PHP       | Philippines Pesos    |      0.022535 |
| SGD       | Singapore Dollars    |      0.743498 |
| USD       | US Dollars           |      1.000000 |
+-----------+----------------------+---------------+
9 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM `group`;
+---------+-------------------+----------+---------------+
| gr_code | gr_name           | gr_limit | gr_limit_curr |
+---------+-------------------+----------+---------------+
| ARRG    | Arrow Group       | 34000000 | USD           |
| HARG    | Harrow Industries | 22000000 | USD           |
| MTKG    | Martek Group      |  6000000 | USD           |
+---------+-------------------+----------+---------------+
3 rows in set (0.00 sec)

mysql> SELECT * FROM industry;
+---------+-------------+
| ic_code | ic_name     |
+---------+-------------+
|       1 | Automotive  |
|       2 | Mining      |
|       3 | Textile     |
|       4 | Chemical    |
|       5 | Electronics |
+---------+-------------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM officer;
+--------+---------------+-------------------+---------+------------+
| off_id | off_name      | position          | br_code | supervisor |
+--------+---------------+-------------------+---------+------------+
|      1 | Marshal, J.T. | CEO               | SYD     |          1 |
|      2 | Henderson, R. | Sydney Manager    | SYD     |          1 |
|      3 | Chen, G.      | Melbourne Manager | MEL     |          1 |
|      4 | Farquhar, M.  | Auckland Manager  | AUC     |          1 |
|      5 | Vijay, R.     | Dealer            | SYD     |          2 |
|      6 | Ick, D.       | Dealer            | SYD     |          2 |
|      7 | Dabinett, B.  | Dealer            | MEL     |          3 |
|      8 | Binny, S.     | Dealer            | MEL     |          3 |
```

```
|      9 | Suffin, L.    | Dealer            | AUC     |          4 |
|     10 | Wiskins, M.   | Dealer            | AUC     |          4 |
+--------+---------------+-------------------+---------+------------+
10 rows in set (0.00 sec)

mysql> SELECT * FROM payment;
+------------+--------+-----------------+--------------+-------------------+
| payment_id | con_id | payment_due_date | payment_amnt | payment_made_date |
+------------+--------+-----------------+--------------+-------------------+
|          1 |      5 | 2009-11-04       |      2000000 | 2009-11-04        |
|          2 |      5 | 2010-06-15       |      1400000 | NULL              |
|          3 |     10 | 2009-09-01       |      3000000 | 2009-09-01        |
|          4 |     10 | 2009-12-01       |      3000000 | 2009-12-01        |
|          5 |     10 | 2010-03-01       |      3000000 | 2010-03-01        |
|          6 |     10 | 2010-06-01       |      4000000 | NULL              |
|          7 |     15 | 2010-08-30       |      2000000 | NULL              |
|          8 |     15 | 2011-04-01       |      1500000 | NULL              |
|          9 |     20 | 2010-04-15       |      5000000 | 2010-04-15        |
|         10 |     20 | 2010-06-30       |      5000000 | NULL              |
|         11 |     20 | 2010-08-01       |     11000000 | NULL              |
|         12 |     25 | 2010-04-18       |      1000000 | 2010-04-18        |
|         13 |     25 | 2010-12-01       |      8500000 | NULL              |
|         14 |     30 | 2010-08-01       |      6000000 | NULL              |
|         15 |     30 | 2011-01-15       |      6000000 | NULL              |
|         16 |     40 | 2010-06-12       |      1500000 | NULL              |
|         17 |     40 | 2010-08-08       |       600000 | NULL              |
|         18 |     50 | 2011-04-20       |      2000000 | NULL              |
|         19 |     50 | 2011-10-01       |      3000000 | NULL              |
|         20 |     60 | 2010-10-10       |      3200000 | NULL              |
+------------+--------+-----------------+--------------+-------------------+
20 rows in set (0.00 sec)
```

Task 3

Provide the SQL (and output) that lists the names of all branches that have dealers, and the number of dealers working for each of these branches.

```
mysql> SELECT b.br_name, COUNT( o.off_id ) AS dealer_count
    -> FROM branch b
    -> INNER JOIN officer o ON o.br_code = b.br_code
    -> AND o.position = 'Dealer'
    -> GROUP BY b.br_code;
+-----------+--------------+
| br_name   | dealer_count |
+-----------+--------------+
| Auckland  |            2 |
| Melbourne |            2 |
| Sydney    |            2 |
+-----------+--------------+
3 rows in set (0.01 sec)
```

Task 4

Provide the SQL (and output) that lists the names of officers who have not made loans at branches other than their home branch.

```
mysql> SELECT o.off_name
    -> FROM officer o
    -> WHERE o.off_id NOT
    -> IN (
    -> SELECT o.off_id
    -> FROM contract con
    -> INNER JOIN officer o ON o.off_id = con.off_id
    -> AND con.br_code != o.br_code
    -> );
+---------------+
| off_name      |
+---------------+
| Marshal, J.T. |
| Henderson, R. |
| Chen, G.      |
| Farquhar, M.  |
| Ick, D.       |
| Dabinett, B.  |
| Binny, S.     |
| Suffin, L.    |
+---------------+
8 rows in set (0.00 sec)
```

Task 5

Provide the SQL (and output) that determines the amount in USD that the bank would expect to receive in payments during the third quarter 2010, given the current loans portfolio.

```
mysql> SELECT ROUND(SUM( p.payment_amnt *
    -> curr.usd_xchg_rate )) AS expected_amount
    -> FROM contract con
    -> INNER JOIN payment p ON p.con_id = con.con_id
    -> INNER JOIN currency curr ON curr.curr_code = con.curr_code
    -> WHERE (
    -> p.payment_due_date
    -> BETWEEN '2010-07-01'
    -> AND '2010-11-30');

+-----------------+
| expected_amount |
+-----------------+
|        19988305 |
+-----------------+
1 row in set (0.00 sec)
```

Task 6

Using a subquery (and join) find the name of the client with the loan that involves the largest number of payments, and the number of payments for that loan.

```
mysql> SELECT c.cl_name, con.con_id,
    -> COUNT( p.con_id ) AS  payment_count
    -> FROM client c
    -> INNER JOIN contract con ON con.cl_code = c.cl_code
    -> INNER JOIN payment p ON p.con_id = con.con_id
    -> GROUP BY c.cl_code, con.con_id
    -> ORDER BY payment_count DESC
    -> LIMIT 1;
+---------------------+--------+---------------+
| cl_name             | con_id | payment_count |
+---------------------+--------+---------------+
| Arrow International |     10 |             4 |
+---------------------+--------+---------------+
1 row in set (0.00 sec)
```

Task 7

Provide the SQL (and output) that determines which industries (that are listed in the database) are not currently borrowing from the Bank.

```
mysql> SELECT *
    -> FROM industry
    -> WHERE ic_code NOT
    -> IN (
    -> SELECT i.ic_code
    -> FROM industry i
    -> INNER JOIN client c ON c.ic_code = i.ic_code
    -> INNER JOIN contract con ON con.cl_code = c.cl_code
    -> );
+---------+----------+
| ic_code | ic_name  |
+---------+----------+
|       4 | Chemical |
+---------+----------+
1 row in set (0.00 sec)
```

Task 8

Create a view that shows only large loans, namely those whose total payments exceed $USD5 million. Your view should show client name, contract date, payment total and number of payments. List what is seen through this view.

```
mysql> CREATE OR REPLACE VIEW large_loans AS
    -> SELECT c.cl_name, con.con_date, ROUND(SUM(p.payment_amnt *
    -> curr.usd_xchg_rate)) AS payment_total,
    -> COUNT(p.con_id) AS payment_count
    -> FROM client c
    -> INNER JOIN contract con ON con.cl_code = c.cl_code
    -> INNER JOIN currency curr ON curr.curr_code = con.curr_code
    -> INNER JOIN payment p ON p.con_id = con.con_id
    -> GROUP BY p.con_id
    -> HAVING (
    -> payment_total >5000000
    -> )
    -> ORDER BY payment_total DESC;
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT * FROM large_loans;
+--------------------+------------+---------------+---------------+
| cl_name            | con_date   | payment_total | payment_count |
+--------------------+------------+---------------+---------------+
| Arrow Holdings Ltd | 2010-02-15 |      19211703 |             3 |
| Arrow International | 2009-06-01 |       9348508 |             4 |
| Arrow International | 2010-04-12 |       8629392 |             2 |
| Harrow (NZ) Ltd    | 2010-02-18 |       7063231 |             2 |
+--------------------+------------+---------------+---------------+
4 rows in set (0.00 sec)
```

Task 9

For all subordinates of any specified branch manager provide the name of the subordinate officer and the number of loans made by that officer. Provide the manager ID at run time (perhaps by using a query *placeholder*). Demonstrate for the manager with ID 03 (Chen).

```
mysql> SELECT o.off_name, COUNT( con.con_id ) As loan_count
    -> FROM officer o
    -> INNER JOIN contract con ON con.off_id = o.off_id
    -> WHERE o.supervisor = 3
    -> GROUP BY con.off_id;
+-----------+------------+
| off_name  | loan_count |
+-----------+------------+
| Binny, S. |          2 |
+-----------+------------+
1 row in set (0.00 sec)
```

Task 10

Provide the SQL (and output) that can generate the Bank's USD exposure to any nominated group. Again, specify the group ID at run time. Your result should show the group name and the current exposure in USD. Demonstrate using the Arrow group.

```
mysql> SELECT g.gr_name, ROUND( SUM( c.cl_limit *
    -> curr.usd_xchg_rate )) AS current_exposure
    -> FROM `group` g
    -> INNER JOIN client c ON c.gr_code = g.gr_code
    -> INNER JOIN currency curr ON curr.curr_code = c.cl_limit_curr
    -> AND g.gr_code = 'ARRG'
    -> GROUP BY c.gr_code;
+-------------+------------------+
| gr_name     | current_exposure |
+-------------+------------------+
| Arrow Group |         45000000 |
+-------------+------------------+
1 row in set (0.00 sec)
```

Task 11

Define "Availability" as the amount a client may still borrow, that is the client's limit minus the current exposure. You are required to write the SQL code that will produce a "Client Availability Report" which is specified as follows: For every client in the database, list (in client name order) the client's name, the client's limit, and the availability for that client in USD. Provide totals for both the client limit and availability columns.

```
mysql> SELECT c.cl_name, c.cl_limit,
    -> ROUND((c.cl_limit * c_curr.usd_xchg_rate) -
    -> SUM(con.con_amnt * curr.usd_xchg_rate)) AS availability
    -> FROM client c
    -> INNER JOIN contract con ON con.cl_code = c.cl_code
    -> INNER JOIN currency curr ON curr.curr_code = con.curr_code
    -> INNER JOIN currency c_curr
    -> ON c_curr.curr_code = c.cl_limit_curr
    -> GROUP BY con.cl_code;
+----------------------+----------+--------------+
| cl_name              | cl_limit | availability |
+----------------------+----------+--------------+
| Aust Electronics Inc | 12000000 |      5922200 |
| Arrow Holdings Ltd   | 15000000 |     -3296860 |
| Arrow International  | 10000000 |     -6539668 |
| Differentials Inc    |  4000000 |      1769506 |
| Harrow (NZ) Ltd      | 12000000 |      5308518 |
| Martek Mining Inc    |  5000000 |      2842652 |
| Martek Exploration   |  3000000 |     -1000000 |
+----------------------+----------+--------------+
7 rows in set (0.00 sec)
```

Database SQL Dump

```
CREATE DATABASE `currency_loans`;

USE `currency_loans`;

-- --------------------------------------------------------

--

-- Table structure for table `branch`

--

CREATE TABLE IF NOT EXISTS `branch` (

  `br_code` char(3) NOT NULL,

  `br_name` varchar(25) NOT NULL,

  PRIMARY KEY (`br_code`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `branch`

--

INSERT INTO `branch` (`br_code`, `br_name`) VALUES

('AUC', 'Auckland'),

('BRI', 'Brisbane'),

('MEL', 'Melbourne'),

('SYD', 'Sydney'),

('WEL', 'Wellington');

-- --------------------------------------------------------
```

```
--

-- Table structure for table `client`

--

CREATE TABLE IF NOT EXISTS `client` (

  `cl_code` varchar(5) NOT NULL,

  `cl_name` varchar(25) NOT NULL,

  `ic_code` int(11) NOT NULL,

  `cl_limit` int(10) NOT NULL,

  `cl_limit_curr` char(3) NOT NULL,

  `gr_code` char(4) DEFAULT NULL,

  PRIMARY KEY (`cl_code`),

  KEY `ic_code` (`ic_code`,`cl_limit_curr`,`gr_code`),

  KEY `cl_limit_curr` (`cl_limit_curr`),

  KEY `gr_code` (`gr_code`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `client`

--

INSERT INTO `client` (`cl_code`, `cl_name`, `ic_code`, `cl_limit`, `cl_limit_curr`, `gr_code`)
VALUES

('AEI', 'Aust Electronics Inc', 5, 12000000, 'USD', NULL),

('ARH', 'Arrow Holdings Ltd', 5, 15000000, 'USD', 'ARRG'),

('ARI', 'Arrow International', 5, 10000000, 'USD', 'ARRG'),

('ARR', 'Arrow Ltd', 5, 20000000, 'USD', 'ARRG'),

('BCC', 'Barton Chemical Inc', 4, 40000000, 'USD', NULL),

('DIFF', 'Differentials Inc', 2, 4000000, 'USD', NULL),

('HAR', 'Harrow (NZ) Ltd', 3, 12000000, 'USD', 'HARG'),

('HARA', 'Harrow Australasia', 3, 15000000, 'USD', 'HARG'),

('MTK', 'Martek Mining Inc', 1, 5000000, 'USD', 'MTKG'),

('MTKE', 'Martek Exploration', 1, 3000000, 'USD', 'MTKG'),

('NZT', 'New Zealand Tyres', 2, 4000000, 'USD', NULL);

-- --------------------------------------------------------
```

```
--

-- Table structure for table `contract`

--

CREATE TABLE IF NOT EXISTS `contract` (

  `con_id` int(11) NOT NULL,

  `con_date` date NOT NULL,

  `cl_code` varchar(5) NOT NULL,

  `off_id` int(11) NOT NULL,

  `br_code` varchar(3) NOT NULL,

  `con_amnt` int(10) NOT NULL,

  `curr_code` char(3) NOT NULL,

  PRIMARY KEY (`con_id`),

  KEY `cl_code` (`cl_code`),

  KEY `off_id` (`off_id`),

  KEY `br_code` (`br_code`),

  KEY `curr_code` (`curr_code`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `contract`

--

INSERT INTO `contract` (`con_id`, `con_date`, `cl_code`, `off_id`, `br_code`, `con_amnt`,
`curr_code`) VALUES

(5, '2009-04-12', 'MTK', 10, 'SYD', 3000000, 'NZD'),

(10, '2009-06-01', 'ARI', 5, 'SYD', 12000000, 'NZD'),

(15, '2009-12-04', 'DIFF', 8, 'MEL', 3000000, 'SGD'),

(20, '2010-02-15', 'ARH', 5, 'MEL', 20000000, 'AUD'),

(25, '2010-02-18', 'HAR', 9, 'AUC', 9000000, 'SGD'),

(30, '2010-04-12', 'ARI', 6, 'SYD', 11000000, 'NZD'),

(40, '2010-04-15', 'AEI', 8, 'MEL', 2000000, 'GBP'),

(50, '2010-04-20', 'MTKE', 10, 'AUC', 4000000, 'USD'),

(60, '2010-04-28', 'AEI', 6, 'SYD', 3000000, 'USD');

------------------------------------------------------------
```

```
--
-- Table structure for table `currency`
--
CREATE TABLE IF NOT EXISTS `currency` (
  `curr_code` char(3) NOT NULL,
  `curr_name` varchar(25) NOT NULL,
  `usd_xchg_rate` decimal(10,6) NOT NULL,
  PRIMARY KEY (`curr_code`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Dumping data for table `currency`
--
INSERT INTO `currency` (`curr_code`, `curr_name`, `usd_xchg_rate`) VALUES
('AUD', 'Australian Dollars', '0.914843'),
('EUR', 'Euros', '1.272300'),
('GBP', 'United Kingdom Pounds', '1.538900'),
('JPY', 'Japanese Yen', '0.011966'),
('NOK', 'Norwegian Krone ', '0.161349'),
('NZD', 'New Zealand Dollars', '0.719116'),
('PHP', 'Philippines Pesos', '0.022535'),
('SGD', 'Singapore Dollars', '0.743498'),
('USD', 'US Dollars', '1.000000');
-- --------------------------------------------------------
```

```
--
-- Table structure for table `group`
--
CREATE TABLE IF NOT EXISTS `group` (
  `gr_code` char(4) NOT NULL,
  `gr_name` varchar(25) NOT NULL,
  `gr_limit` int(10) NOT NULL,
  `gr_limit_curr` char(3) NOT NULL,
  PRIMARY KEY (`gr_code`),
  KEY `gr_limit_curr` (`gr_limit_curr`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Dumping data for table `group`
--
INSERT INTO `group` (`gr_code`, `gr_name`, `gr_limit`, `gr_limit_curr`) VALUES
('ARRG', 'Arrow Group', 34000000, 'USD'),
('HARG', 'Harrow Industries', 22000000, 'USD'),
('MTKG', 'Martek Group', 6000000, 'USD');
-- --------------------------------------------------------
```

```
--

-- Table structure for table `industry`

--

CREATE TABLE IF NOT EXISTS `industry` (

  `ic_code` int(11) NOT NULL AUTO_INCREMENT,

  `ic_name` varchar(15) NOT NULL,

  PRIMARY KEY (`ic_code`)

) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=6 ;


--

-- Dumping data for table `industry`

--

INSERT INTO `industry` (`ic_code`, `ic_name`) VALUES

(1, 'Automotive'),

(2, 'Mining'),

(3, 'Textile'),

(4, 'Chemical'),

(5, 'Electronics');


-- --------------------------------------------------------

--

-- Table structure for table `officer`

--

CREATE TABLE IF NOT EXISTS `officer` (

  `off_id` int(11) NOT NULL AUTO_INCREMENT,

  `off_name` varchar(25) NOT NULL,

  `position` varchar(25) NOT NULL,

  `br_code` char(3) NOT NULL,

  `supervisor` int(11) NOT NULL,

  PRIMARY KEY (`off_id`),

  KEY `br_code` (`br_code`,`supervisor`),

  KEY `supervisor` (`supervisor`)

) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=11 ;
```

```
--

-- Dumping data for table `officer`

--


INSERT INTO `officer` (`off_id`, `off_name`, `position`, `br_code`, `supervisor`) VALUES

(1, 'Marshal, J.T.', 'CEO', 'SYD', 1),

(2, 'Henderson, R.', 'Sydney Manager', 'SYD', 1),

(3, 'Chen, G.', 'Melbourne Manager', 'MEL', 1),

(4, 'Farquhar, M.', 'Auckland Manager', 'AUC', 1),

(5, 'Vijay, R.', 'Dealer', 'SYD', 2),

(6, 'Ick, D.', 'Dealer', 'SYD', 2),

(7, 'Dabinett, B.', 'Dealer', 'MEL', 3),

(8, 'Binny, S.', 'Dealer', 'MEL', 3),

(9, 'Suffin, L.', 'Dealer', 'AUC', 4),

(10, 'Wiskins, M.', 'Dealer', 'AUC', 4);


-- --------------------------------------------------------

--

-- Table structure for table `payment`

--

CREATE TABLE IF NOT EXISTS `payment` (

  `payment_id` int(11) NOT NULL AUTO_INCREMENT,

  `con_id` int(11) NOT NULL,

  `payment_due_date` date NOT NULL,

  `payment_amnt` int(10) NOT NULL,

  `payment_made_date` date DEFAULT NULL,

  PRIMARY KEY (`payment_id`),

  KEY `con_id` (`con_id`),

  KEY `con_id_2` (`con_id`)

) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=21 ;
```

```
--

-- Dumping data for table `payment`

--

INSERT INTO `payment` (`payment_id`, `con_id`, `payment_due_date`, `payment_amnt`,
`payment_made_date`) VALUES

(1, 5, '2009-11-04', 2000000, '2009-11-04'),

(2, 5, '2010-06-15', 1400000, NULL),

(3, 10, '2009-09-01', 3000000, '2009-09-01'),

(4, 10, '2009-12-01', 3000000, '2009-12-01'),

(5, 10, '2010-03-01', 3000000, '2010-03-01'),

(6, 10, '2010-06-01', 4000000, NULL),

(7, 15, '2010-08-30', 2000000, NULL),

(8, 15, '2011-04-01', 1500000, NULL),

(9, 20, '2010-04-15', 5000000, '2010-04-15'),

(10, 20, '2010-06-30', 5000000, NULL),

(11, 20, '2010-08-01', 11000000, NULL),

(12, 25, '2010-04-18', 1000000, '2010-04-18'),

(13, 25, '2010-12-01', 8500000, NULL),

(14, 30, '2010-08-01', 6000000, NULL),

(15, 30, '2011-01-15', 6000000, NULL),

(16, 40, '2010-06-12', 1500000, NULL),

(17, 40, '2010-08-08', 600000, NULL),

(18, 50, '2011-04-20', 2000000, NULL),

(19, 50, '2011-10-01', 3000000, NULL),

(20, 60, '2010-10-10', 3200000, NULL);


-- --------------------------------------------------------

--

-- Constraints for table `client`

--

ALTER TABLE `client`

  ADD CONSTRAINT `client_ibfk_1` FOREIGN KEY (`ic_code`) REFERENCES `industry` (`ic_code`) ON
DELETE NO ACTION ON UPDATE NO ACTION,

  ADD CONSTRAINT `client_ibfk_2` FOREIGN KEY (`cl_limit_curr`) REFERENCES `currency`
(`curr_code`) ON DELETE NO ACTION ON UPDATE NO ACTION,

  ADD CONSTRAINT `client_ibfk_3` FOREIGN KEY (`gr_code`) REFERENCES `group` (`gr_code`) ON
DELETE NO ACTION ON UPDATE NO ACTION;
```

```
--

-- Constraints for table `contract`

--

ALTER TABLE `contract`

  ADD CONSTRAINT `contract_ibfk_1` FOREIGN KEY (`cl_code`) REFERENCES `client` (`cl_code`) ON
DELETE NO ACTION ON UPDATE NO ACTION,

  ADD CONSTRAINT `contract_ibfk_2` FOREIGN KEY (`off_id`) REFERENCES `officer` (`off_id`) ON
DELETE NO ACTION ON UPDATE NO ACTION,

  ADD CONSTRAINT `contract_ibfk_3` FOREIGN KEY (`br_code`) REFERENCES `branch` (`br_code`) ON
DELETE NO ACTION ON UPDATE NO ACTION,

  ADD CONSTRAINT `contract_ibfk_4` FOREIGN KEY (`curr_code`) REFERENCES `currency`
(`curr_code`) ON DELETE NO ACTION ON UPDATE NO ACTION;



--

-- Constraints for table `group`

--

ALTER TABLE `group`

  ADD CONSTRAINT `group_ibfk_1` FOREIGN KEY (`gr_limit_curr`) REFERENCES `currency`
(`curr_code`) ON DELETE NO ACTION ON UPDATE NO ACTION;



--

-- Constraints for table `officer`

--

ALTER TABLE `officer`

  ADD CONSTRAINT `officer_ibfk_1` FOREIGN KEY (`br_code`) REFERENCES `branch` (`br_code`) ON
DELETE NO ACTION ON UPDATE NO ACTION,

  ADD CONSTRAINT `officer_ibfk_2` FOREIGN KEY (`supervisor`) REFERENCES `officer` (`off_id`)
ON DELETE NO ACTION ON UPDATE NO ACTION;



--

-- Constraints for table `payment`

--

ALTER TABLE `payment`

  ADD CONSTRAINT `payment_ibfk_1` FOREIGN KEY (`con_id`) REFERENCES `contract` (`con_id`) ON
DELETE NO ACTION ON UPDATE NO ACTION;
```