

GENERAL COMMENTS – CI204 – GENERAL COMMENTS

Please review, from the assessment, what you are to evidence (from the assignment). **Additional comments in red.** Also, review the marking rubric (in *Module -> Assessment* tab).

5) Create tables using SQL DDL Write SQL statements to create the tables that implement the database you designed in part 1. Your report must show the SQL CREATE TABLE statements and images of the tables created in SQL Server or MySQL or another DBMS approved by the module tutor. **[Use annotated screenshots as evidence. See Appendix A for example annotated screenshot example].**

6) Populate tables using SQL INSERT Write SQL statements to populate all the tables you have previously created, with at least five records each. Your report must show the SQL INSERT statements and images of the tables when they have been populated. *NOTE: you should consider creating a formal test data set where you know what the outcome of a given SQL query will be* **[See Appendix B for an example of a formal test set]**

7) Retrieve information using SQL queries Write SQL queries to demonstrate that your system can address the requirements in Appendix A. Your report must show the SQL code and the result of executing each query. These should be clearly labelled with the requirement number. **[This will be assessed against each query being demonstrated. Use annotated screenshots as evidence. See Appendix A for example annotated screenshot example].**

8) Implementation of DBMS functionality Design appropriate stored procedures, functions, triggers and application to implement the functionality defined in Appendix A. Your report must show the SQL code and the result of execution.

COMMENTS FROM PAST YEARS' - SIMILAR ASSIGNMENTS – DIFFERENT CASE STUDY

- SQL in screen prints are not legible – consider pasting SQL underneath
- Too much white space in screenshots
- Not all tables used in later queries are shown being created / populated
- (Creating tables) Tables / fields / datatypes not shown in screenshots (i.e. tabs not expanded in SQL Server)
- The SQL that you have shown is clearly machine-generated – you were asked to demonstrate your own SQL skills
- (Creating tables) Need constraints around compound PKs to ensure uniqueness
- (Creating tables) No evidence / testing of constraints shown
- You have populated tables at the many end of a relationship before populating the one end of the relationship – this should fail according to your constraints
- (Creating tables) No foreign key SQL shown
- Can't see what results should be – use sample test data with clear outcomes expected
- (Adding / Deleting records) – if you are populating multiple tables then you will need to confine within a transaction, so that if one fails, all fail e.g. booking / staff
- Demonstrate that SQL has deleted records (see transaction point above if across multiple tables)
- Not all expected fields shown – e.g. to create mail-merge to email all members of the same group, would expect to have their names as well as email addresses / address of event that you are discussing
- Have shown how to add and delete records, but not how to retrieve them.
- Delete statements should have evidence of before and after (to show that they work)
- No evidence of execution of query – just SQL which may, or may not, work
- Queries not referenced using numbers from assignment, making it hard to work what each part is evidence for
- All figures should be titled, with reference to which requirement they are meeting
- Use of Cartesian product, rather than joins, across tables
- Order of results not suitable for defined task (various reasons)
- Stored procedure created, but used hard-coded values, not input parameters
- Trigger created, but no evidence shown of it working

APPENDIX A: Example of annotated screenshot (for a library case study)

Query One – All current loans

Evidence of 4 tables in database, showing field names and data types

Note:

- no wasted white space (have changed window size before screenshot)
- SQL underneath in text to aid readability
- Results as expected (see Appendix B for data with markups).

```
select tMember.memberNo
, surname
, forename
, tBook.bookNo
, bookTitle
from tMember
inner join tLoan
on tMember.memberNo = tLoan.memberNo
inner join tBook
on tBook.bookNo = tLoan.bookNo
and ret = 0
order by tMember.memberNo
```

	memberNo	surname	forename	bookNo	bookTitle
1	1001	Dodd	Elise	200	The SQL papers
2	1003	Hooper	Gill	400	A guide to INGRES SQL
3	1004	Worth	Nichola	200	The SQL papers
4	1004	Worth	Nichola	1200	Data management and file structures

Query executed successfully. | cssql (14.0 RTM) | UNIVERSITY\jh1033 (55) | jh1033_3 | 00:00:00 | 4 rows

```
select tMember.memberNo, surname, forename, tBook.bookNo, bookTitle from tMember
inner join tLoan
on tMember.memberNo = tLoan.memberNo
inner join tBook
on tBook.bookNo = tLoan.bookNo
and ret = 0
order by tMember.memberNo
```

APPENDIX B: Example of a test data set (for a library case study)

Query One – All current loans
Query Two – All loans for one member
Query Three – For each title (not copy) - total times loaned out

Test data listings indicating query results

tClass

classCode	className
005.756	Programming : relational databases
005.7565	Programming : specific relational databases

tBook – shaded records indicate no loans for that book

bookNo	bookTitle	author	classCode
100	Database for the IBM PC with SQL	Emerson Sandra	005.7565
200	The SQL papers	Stonebraker Michael	005.7565
300	A guide to INGRES SQL	Date Chris	005.7565
400	A guide to INGRES SQL	Date Chris	005.7565
500	Analysis and comparison of relational databases	Valduriez Patrick	005.756
600	Analysis and comparison of relational databases	Valduriez Patrick	005.756
700	Handbook of relational database design	Fleming C	005.756
800	Handbook of relational database design	Fleming C	005.756
900	Object-oriented database programming	Alagic Suad	005.756
1000	Object-oriented database programming	Alagic Suad	005.756
1100	Data management and file structures	Loomis Mary	005.756
1200	Data management and file structures	Loomis Mary	005.756

tMember – shaded records indicate no loans for that member

memberNo	surname	forename	telNo	deptNo	mbr_fines
1001	Dodd	Elise	01273-699699	1	0.00
1002	Tubing	Akram	01273-556699	6	22.30
1003	Hooper	Gill	07729-444321	3	1.20
1004	Worth	Nichola	020-8546-258	1	0.00
1005	Al Khowarizmi	Mohammed	020-8975-669	5	1.45
1006	Von Neeman	Julia	07880-565621	1	0.35
1007	Mendelbrat	Anna	020-75648-236	6	0.00
1008	Epstein	Astrid	020-74569-632	1	2.00
1009	Bierpinski	Stella	020-78877-665	4	53.00

tLoan - shaded records show current loans

bookNo	memberNo	dueBack	ret
200	1003	2019-06-23	1
200	1004	2019-05-12	0
200	1001	2019-05-14	0
300	1002	2019-05-06	1
400	1003	2019-09-03	0
900	1005	2019-09-18	1
1200	1004	2019-06-14	0