**Task 1**

First thing to check for is that they have produced a class diagram and not an ER diagram

The scenarios should not be too complex or simple, I’d be looking for between 4-6 classes, but it must represent a case study requiring complex objects,

there should relevant inheritance and some clear way that they can use nested tables and/or arrays for aggregation.

Entity names should be singular, and I would be expecting association relationships to be labelled and with cardinality/optionality clearly shown.

Attributes should be shown on the diagram, ideally with types.

The diagram doesn’t need to show methods but don’t penalise them if they do.

**Task 2**

What I expect to see in task 2 is an **object-relational** implementation of the UML diagram. So, if using Oracle use of types for inheritance, incorporation of nested tables and/or arrays for aggregation where relevant.

ensure tables are dropped (normally in reverse order) before they are created.

You should create types, then tables (using types where relevant).

Also check attribute data types to check they are sensible (e.g. suitable sizes/precision for the data required), and that they are using varchar for data such as telephone numbers which can start with 0.

Also include use of a timestamp/date/interval which you can then use in task 4.

Also, it’s better to use NUMBER/NUMERIC than INT so they can specific precision (check that the precision stated is sensible: eg Number(2,0), i.e. they don’t just use the same size for each attribute).

Check they have sufficient data, not looking for a lot of data, a few rows for each table, and that for example numeric data isn’t in quotes, and they’ve used null, not ‘’. They also need some sort of numeric data so that they can use this in their OLAP query in task 4 – so check that they have something suitable for this. Also, if they have data such as an address, it should be split into separate fields (street, town, etc.) or preferably they could use a complex type.

They should provide evidence that their script works in the PDF document they provide, and they should also provide a complete script file in a .sql file. Check that they that have written the code themselves and it’s not code that’s been automated by Oracle/PostgreSQL. If so, then the max mark is Major issues.

**Task 3**

There are two important things to check for here.

Firstly they are required to provide all the MongoDB commands (e.g. db.collectionname.insert), rather than a JSON output of their database from MongoDB. Reduce their mark by one criteria if they don’t give all their MongoDB code. If the screenshots in the report show all the code then that’s fine.

Secondly, and probably most importantly, it should be a **document store** implementation. Many students don’t understand this and simply replicate their relational design, i.e. one collection per table.

I am looking for them to have appropriately used nested documents, arrays, and documents with different properties in the same collection (to show they understand that it’s a semi-structured schema), and also removes the need to store null data. They should also be using the ISODate or Date data type for any dates they insert so that they can use this in task 4..

**Task 4/5**

Last year queries 1, 3 and 4 are the same, but query 2 last year was to use a subtype whereas this year they should use a nested table or inheritance (depending on what they implemented in task 2). Check that query b is not a subquery.

**Task 4a/5a**

In **Oracle/PostgreSQL**: I’m looking for a query involving at least three tables. To get full marks they could include more tables and need to include both outer (left/right or full) and inner joins. It should also include some sort of restriction on the data. If they don’t have a restriction or don’t use different types of joins then minor issues. Clearly for each task I’m giving higher marks based on complexity of this query so take this into account when marking. It should be beyond a simple query with only inner joins, and of course the query must have some meaning.

When developing an equivalent in **MongoDB**, they get full marks if they write a query which returns the same data. They have been taught how to use DBRef, how to use ObjectIDs and how to use the aggregation pipeline for performing queries, so I’d expect to see them using this. Look for them to have used a nested document, which reduces the need for joins in MongoDB, and this clearly shows good understanding of MongoDB so this would get full marks. Make sure in all MongoDB queries that they don’t just perform a simple search to return the same data, but that it doesn’t match the SQL.

If they submit SQL code for their MongoDB query then no marks as although this is possible, I have specifically said they need to use Mongo query language, not SQL.

**Task 4b/5b**

For this one I expect to see a query which either uses inheritance or a nested table. If using PostgreSQL then using table inheritance.

Again, higher marks the more complex, e.g. if looking at subtypes then being able to retrieve/query based on specific columns within the subtype rather than just retrieving all data, for example.

In MongoDB I’d ideally like to see a query which uses a nested document. Or if they’ve properly used different properties for documents in the same collection, then something simple which demonstrates this. So actually the MongoDB query could be fairly simple.

Many students just provide a subquery, this is incorrect as it does not meet the requirements.

**Task 4c/5c**

In **Oracle/PostgreSQL**, I want to see a query either using intervals, or using for example the extract feature to retrieve portions of dates. Many just give a query using, for example, months\_between, which is not at a suitable level as they did this in year 2. Also just comparing dates is not at the appropriate level of challenge.

They should include joins/restrictions to include detailed data, etc., to get full marks. This applies for all queries.

For **MongoDB¸** they may struggle to replicate what they do in Oracle, but they can use ISODate or Date functions to extract portions of dates.

**Task 4d/5d**

Full marks if they use partition, including one/more of sum, max, avg, etc. ‘Rollup/cube ‘meets expected standard’, just make a comment that they lack challenge. If you only get a group by then this is minor issues maximum, depending on complexity, as they did group by in 2nd year.

The rollup/cube must be over multiple dimensions (i.e. two or more columns) otherwise it’s pointless. Make sure that the query has some sensible meaning.

They should include joins/restrictions to get full marks.

They will struggle to replicate this in MongoDB, so Full marks if they do – ‘meets expected standard’ if they come close. They have been given code which replicates rollup (they weren’t given this last year), so they should get close.

**Task 6 Discussion**

For each query I want a brief comparison. They should be highlighting what they have learned, the challenges they encountered, especially in translating from SQL to MongoDB, and which of Oracle/MongoDB they prefer and why.

Each discussion should relate back to the example queries and they should be using references, correctly in Harvard format. One or two references for each discussion preferably.

They should not be talking about tables when referencing MongoDB, which uses collections of documents.