

Assignment 1 (10% of total marks)

Due date: Sunday, 31 October, 2021 by 9:00 pm SGT.

Scope:

The tasks of this assignment cover **functional dependency and indexing**. The assignment covers the topics discussed in lecture 1, 2, and 3.

Assessment criteria:

Marks will be awarded for:

- Correct,
- Comprehensive, and
- Appropriate

application of the materials covered in this subject.

The specification of the assignment starts from next page.

$$\begin{array}{l} \downarrow C, D, B \\ A \rightarrow B, E, F \\ \downarrow b \end{array}$$

Assignment Specification:

Task 1 (3.0 marks)

Analysis of relational schemas and normalization

A book is described by a catalog number, a book title, author's first name, author's last name, a publisher name, the year the book was published, and a price of the book. An author may write many books and each book may be written by one or more authors. The information is stored in a relational table book as follow:

BOOK (CatalogNum, BookTitle, AuthorFName, AuthorLName, PublisherName, YearOfPublication, Price)

The following dependencies exist in the relational table BOOK:

- $CatalogNum \rightarrow BookTitle, PublisherName, YearOfPublication$ (A) $\rightarrow B, E, F$
- $BookTitle, PublisherName, YearOfPublication \rightarrow Price$ (B, E, F) $\rightarrow G$
- $AuthorFName, AuthorLName, BookTitle \rightarrow CatalogNum$ (C, D, B) $\rightarrow A$
Augment to SK

- Find all the minimal super key of the relational table BOOK. List the derivations of all minimal keys.
- Identify the highest normal form of the relational table BOOK. List the justifications for each highest normal form found.
- Decompose the relational table BOOK into minimal number of relational tables in BCNF. List all relational tables obtained from the decompositions.

Deliverables

A file solution1.pdf with the outcomes of the steps (i), (ii), and (iii) listed above. Note, that "educated guesses" of the solutions score no marks. You must provide the complete justifications of your answers.

Submission of a file with a different name and/or different extension and/or different type scores no marks!

B* TREE

Task 2 (4.0 marks)

Indexing

Using the relational table LINEITEM of the sample database TPCHR, for each one of the queries listed below:

- i. Find all the discount (**l_discount**) of all the items that are shipped (**l_shipdate**) most recently. Hint. Most recently mean the latest shipment date.
 - ii. Find the **total number of items** shipped by air (**l_shipmode**) in 1998 (**l_shipdate**). *select count * from lineitems where l_shipmode = 'AIR' and to_char(l_shipdate) = '1998'; Ans: 29684*
 - iii. Find the order number (**l_orderkey**) and item number (**l_linenum**) that have the highest discount (**l_discount**).
 - iv. Find the total number of item per line status (**l_linestatus**). List the line status and the total items per line status.
 - v. Find the order key (**l_orderkey**), line item number (**l_linenum**), line status (**l_linestatus**), shipment date (**l_shipdate**) and shipment mode (**l_shipmode**) of all orders with the order number (**l_orderkey**) 1795718, 1799046, and 1794626.
- a) Construct an SQL statement that produces the required output specified in the statement. **(1.5 marks)**
- b) Find the best possible indexing of a relational table LINEITEM. The best possible indexing means that a database system will compute a query with an index proposed by you using the **smallest number of read block operations**. Note that you can create only one index per query, and there is no need to find indexes that speed up the processing of more than one query. Use the explain plan and show plan statements to justify your solutions. **(2.5 marks)**

one index per query - *→ least # of cost*

Deliverables

A file solution3.pdf with CREATE INDEX statements that improve the performance of the queries listed (i, ii, iii, iv, and v above) and the execution plan generated.

Please remember that you must consider each one of the queries as an individual case! Please remember that all relational tables are large enough to make full table scans more time consuming than accessing the tables through an index! It means that any solution in which an index is not used for query processing is incorrect.

index full = index is in order
required by query
fast full = no order

Task 3 (3.0 marks) Indexing

Using the relational table ORDERS of the sample database TPCHR and the following index created, construct an appropriate SQL statement and satisfies the following requirement:

PK

create index A1Task3Idx on orders(o_orderdate, o_clerk, o_totalprice);

Find SELECT statements that will use the index in the following ways:

- ① a) Execution of the first SELECT statement must traverse the index vertically and it must not access a relational table ORDERS. (0.6 mark)
Index unique scan
- ② b) Execution of the second SELECT statement must first traverse the index vertically followed by horizontal scan at the leaf level. The retrieval operation must not access a relational table ORDERS. (0.6 mark)
Index unique scan, Index fast full scan, x Table Access By Index ROWID
- ③ c) Execution of the third SELECT statement must traverse the leaf level of the index horizontally and it must not access the relational table ORDERS. (0.6 mark)
- leaf = bottom row of B* TREE.
- select c_name, c_email from customers -
- Index fast full scan.
- ④ d) Execution of the fourth SELECT statement must traverse the index vertically and it must access a relational table ORDERS. (0.6 mark)
Index unique scan, Table Access By Index ROWID.
- ⑤ e) Execution of the fifth SELECT statement must first traverse the index vertically followed by horizontal scan at the leaf level. The retrieval operation must access a relational table ORDERS. (0.6 mark)
Index unique scan, Index fast full scan, Table Access By Index ROWID

Deliverables

A file solution3.pdf with the SELECT statements for each one of the five cases described above.

- index range scan = where last_name LIKE 'A%';
- index unique scan = select * from employee where ID=5; vertical

Submissions

This assignment is due on Sunday, 31 October 2021 by 9:00 pm (21:00 hours) Singapore time.

Submit the files **solution1.pdf**, **solution2.pdf**, and **solutions3.pdf** through Moodle in the following way:

- 1) Zip all the files (Solution1.pdf, solution2.pdf, and solution3.pdf into one zipped folder.)
- 2) Access Moodle at <http://moodle.uowplatform.edu.au/>**
- 3) To login use a Login link located in the right upper corner the Web page or in the middle of the bottom of the Web page
- 4) When successfully logged in, select a site CSCI235 (SP421) Database Systems
- 5) Scroll down to a section Submissions of Assignments
- 6) Click at Submit your Assignment 1 here link.
- 7) Click at a button Add Submission
- 8) Move the zipped file created in Step 1 above into an area provided in Moodle. You can drag and drop files here to add them. You can also use a link *Add...*
- 9) Click at a button Save changes,
- 10) Click at check box to confirm authorship of a submission,
- 11) When you are satisfied, remember to click at a button Submit assignment.

A policy regarding late submissions is included in the subject

outline. Only one submission per student is accepted.

Assignment 1 is an individual assignment and it is expected that all its tasks will be solved individually without any cooperation with the other students. Plagiarism is treated seriously. Students involved will likely receive zero. If you have any doubts, questions, etc. please consult your lecturer or tutor during lab classes or over e-mail.

End of specification