**IST 220 – Introduction to Databases**

**Test 1 Review Sheet**

**Format**

In-class written parts (60~70%): due on <test day> by the end of class

* Multiple choice: concepts
* Short answer: concepts
* Applications: generating result manually

Take-home open-book part (30~40%): due on <next class day> at the beginning of class

* Hands-on Exercises: applications (from Lab 1 thru 3)

**Overview (& General)**

Concepts

Data and information: Digitalized data that is stored and processed with computer programs.

Database and DBMS:

* Database: An organized collection of data
* DBMS (Database Management System): A system software suite designed to manage database systems
* Database Application: An application software system that provides interfaces appropriate for end users with different for end users with different tasks in certain business scenarios.

Client and Server, Software System and Machine (Hardware)

* Client: Programs and Apps
* Server: Programs or Systems
* Client Devices
* Server Machines
* server machine and server instance names used in CSC labs

The CRUD operations: CREATE, RETRIEVE, UPDATE, DELETE

Table schema (represented by **columns**) and content (stored in **rows**)

More on columns (aka attributes):

* column name
* data type
* valid data range

**SELECT-FROM-ORDER BY (Lab 1)**

Concepts

Operators: arithmetic, concatenation (+)

SELECT <Expression>[arithmetic operator]<expression>...

FROM [table\_name]

WHERE [expression];

Column alias

SELECT *column\_name* AS *alias\_name*  
FROM *table\_name;*

Punctuations: Single quote, double quote, square brackets, parenthesis.

Keywords: TOP, DISTINCT, AS, ASC/DESC

TOP : SELECT TOP *number*|*percent* *column\_name(s)*  
FROM *table\_name*WHERE *condition*;

DISTINCT: SELECT DISTINCT *column1*, *column2, ...*  
FROM *table\_name*;

AS: RENAME A COLUMN

ASC/DESC: SELECT \* FROM Customers  
ORDER BY CustomerName ASC; (or DESC)

Applications

Select & display all columns

SELECT (Column Name)

FROM (TABLE)

Select & display a number of columns (limiting data by column) in a given order

SELECT (ColumnName)\*

FROM (TABLE)

WHERE TableName=’tablename’

Select & display values based calculation (or concatenation)

CONCAT(*string1*, *string2*, *....*, *string\_n*)

Display results with more readable column captions

SELECT column1, column2 FROM table1, table2 WHERE column2='value';

Sorting results in ascending or descending order

SELECT \* FROM Customers  
ORDER BY CustomerName ASC; (or DESC)

Remove duplicate rows in result set

|  |  |
| --- | --- |
| 4  5  6  7  8  9 | SELECT [FirstName],      [LastName],      [Country],      COUNT(\*) AS CNT  FROM [SampleDB].[dbo].[Employee]  GROUP BY [FirstName],        [LastName],        [Country]  HAVING COUNT(\*) > 1; |
|  |  |

Display *Top-N* rows only

SELECT TOP number|*percent* column\_name(s)  
FROM table\_nameWHERE condition;

**WHERE (Lab 2)**

Concepts

More operators: comparison, logical (AND, OR, NOT)

Keywords: LIKE, BETWEEN … AND …, IN (…, …), IS

Wildcard characters: %, \_

NULL: not applicable or currently not available

Applications

Limit data by row in the WHERE clause

* for numbers
* for dates (before, after, all dates except)

Compound criteria with

* logical operators
* more convenient (but somewhat limited) ways: btw, in, etc

Pattern match for character strings

Match null or non-null values

**INNER JOIN (Lab 3)**

Concepts

Roles of the attributes (aka as columns)

* PK: identifies
* FK: links
* others (descriptive attr’s): describe

Relationship links and one-to-many (or many-to-one) relationship

* relationship path: a complete path linking related tables needed to provide info in need

Joining condition (vs. other conditions)

Conventional syntax and standard syntax (using [INNER] JOIN … ON)

Qualifying (ambiguous) column name by prefixing table name

* and even schema/database name

Table alias

Inner join types

* equijoin
* non-equijoin
* self-join

Applications

*For this test, joins will be limited to using standard syntax and equality condition only.*

*A different DB with 5 or fewer tables may be used on the test.*

Joining two tables

Joining two copies of the same table

Joining three or more tables (n-1 JOINs needed for joining n tables)

* intermediate tables on the relationship path will be needed in the join even if they don’t appear to provide any attribute to form the result set

**OUTER JOIN**

Concepts

INNER JOIN vs. OUTER JOIN

Outer join types

* LEFT
* RIGHT
* FULL

**STATEMENTS TO STUDY**

LAB1: SELECT, FROM, WHERE, ORDER BY

LAB2: NOT, AND OR () statements

LAB3:= >< <>, IN (A,B,….) ( ) A BETWEEN a AND b, NULL, A>=a AND A <=b, LIKE, % ‘IST’. PK=FK Inv.VendorID=Vend.VendorID