**IST 220 – Introduction to Databases**

**Test 2 Review Sheet**

**Format**

In-class written parts (60~70%): due 11/12 by the end of class

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

Take-home open-book part (30~40%): due 11/14 by midnight

* Hands-on Exercises: applications (from Labs 4 & 5; as well as basic SELECT syntax from Labs 1 thru 3)
* BOOKSTORE DATABASE

**Group Functions and GROUP BY-HAVING (Lab 4)**

Concepts

Function:

* function name, followed parenthesis  
  Substring( … )
* list of arguments (or parameters), each requires a specific type of value  
  ( expression, start\_index, number\_of\_char )  
   varchar or char int int
* return type  
  char

Group functions: returns one value (in one row) for the whole group

GROUP BY clause may be used

* after the WHERE clause
* to break down the dataset by one or more columns
* scalar column(s) listed together with aggregate terms must be in GROUP BY
* MIN MAX AVG COUNT

HAVING clause may be used (group level conditions)

* after the GROUP BY clause
* to specify group level condition: an aggregate term should be used

Order of 6 clauses used in SELECT statement is fixed

Applications

Count( expr ) [return type is int] :: three ways to count

* Count (\*)
* Count ( column )
* Count ( distinct column)

Max( expr ), Min( expr ) [return type is the same as that of expr]

Sum( expr ), Avg( expr ) [return type is float] :: applicable to numeric types only

Use SELECT statement with up to 6 clauses

**Scalar Functions (Lab 5)**

Concepts

**Scalar** function: returns a value for each row in the result set

Each DBMS product comes with a different library

Applications

Scalar functions we learned can be categorized as (visit T-SQL/SQL Server online tutorials at <https://msdn.microsoft.com/en-us/library/ms174318.aspx> for more details.)

String functions

* Left/Right(expr, no\_of\_char), Substring: returns string
* Upper/Lower(expr): returns string in the desired case
* Str(float\_expr, len, decimal\_pls): converts float value into string w/ desired pattern
* Len(expr): returns int value, which is length of the string expression

Date/Time functions

* Year/Month/Day(expr): returns desired date part as int
* DateDiff(datepart ,startdate ,enddate ) : returns int, in the desired time elapse measure
* GetDate(): returns the current date and time from the underlying OS

Logical functions

* IsNull(expr): check it out on *Microsoft Devloper’s Network*  
   <https://msdn.microsoft.com/en-us/library/ms184325.aspx>
* IIF( boolean\_expr, true\_value, false\_value ): returns t\_v or f\_v accordingly

Conversion functions

* Cast(expr AS data\_type(len)): returns string representation of the value in expr

**DB Design**

Concepts

ER Modeling:

* Entity and attribute
* Keys: candidate key, primary key, foreign key
* Relationship and cardinality constraint
* Graphical notations

Applications

Using ERM notations

* Draw an ER diagram for modeling a simple problem with 3 or less entity sets
  + Entity boxes, attributes inside box, and PK underlined
  + Relationship lines with side notations for cardinality constraints

Transforming ER model into related table schema

* When one-to-many relationship is involved, each entity set becomes a table, with
  + Each attribute being a column
  + A primary key assigned to each table
  + Candidate key and not null constrains possible
  + A foreign key added to tables on the many side of a one-to-many relationship by copying PK from table on the one side
* An additional (junction) table is needed for each many-to-many relationship, with
  + FKs copied from both related tables
  + A PK formed by both FKs
* One-to-one is not common:
  + May merge the two entity set into one and become one table, or
  + Follow the process for one-to-many

IN CLASS REVIEW

* GROUP FUNCTIONS:

SELECT

FROM

WHERE

GROUP BY

HAVING

ORDER BY to sort results

Scalar functions, design

* Distinguishing Aggregate and Scalar functions:
  + Aggregate: Don’t care about details, or specific order, summary data
    - SUM
    - AVG
    - MIN
    - MAX
    - COUNT
    - Used to return one value from each group
  + Scalar:
    - getDate()
    - YEARN AND MONTH (date)
    - IIF(test,vt (true), vf(false) – list argument, each terms meaning, data,

SELECT COUNT(\*)

FROM BOOKS

Will count how many records are in the books table

WHERE Cat=’Comp’ (Adjusts it so you find how many are in the computers table)

Can also use group by attributes

**SCALAR WITH AGGREGATE**

SELECT PubID, COUNT(\*) OR you could count by specific columns like COUNT(Category) which will count non null values

FROM BOOKS

WHERE Cat=’COMP’

GROUP BY PubID