SQL Structured Query Language

- Agenda
- Topics
- Quiz
- Useful Links

@vmahawar

Agenda

- Introduction – Stakeholders, data classification, Rows/Columns
- DDL – Data Definition Language
  - CREATE, ALTER, DROP, TRUNCATE
  - CONSTRAINTS, DATA TYPES
- DML – Data Manipulation Language
  - INSERT, UPDATE, DELETE, MERGE
- DCL – Data Control Language
  - GRANT, REVOKE
- DQL – Data Query Langauge
  - SELECT, FETCH, DISTINCT, FILTER, LIKE, WITH GROUP BY, HAVING
- TCL – Transaction Control Language
  - COMMIT, ROLLBACK, SAVEPOINT
- OPERATORS – ARITHMETIC, COMPARISON, LOGICAL
- FUNCTIONS
  - STRING, DATE, AGGREGATE, MATH FUNCTIONS
- VIEWS
- TRIGGERS

DDL – Data Definition Language

DDL – Data Definition Language

- CREATE, ALTER, DROP, TRUNCATE
- CONSTRAINTS, DATA TYPES

CREATE TABLE <TABLE NAME>
(COL1 DATATYPE(SIZE) [CONSTRAINTS],
COL2 DATATYPE(SIZE),
..
);

Example:

CREATE TABLE DEPT
(DEPTNO NUMBER(2),
DNAME VARCHAR2(14),
LOC VARCHAR2(13));

DDL – Data Definition Language

DDL – Data Definition Language

- CREATE, ALTER, DROP, TRUNCATE
- CONSTRAINTS, DATA TYPES

```
ALTER TABLE <TABLE NAME>
ADD | MODIFY | DROP
(COL1    DATATYPE(SIZE)   [CONSTRAINTS],
COL2                DATATYPE(SIZE),
..
);
```

Example:

```
CREATE TABLE DEPT
(DEPTNO NUMBER(2),
DNAME VARCHAR2(14),
LOC VARCHAR2(13) );
```

DDL – Data Definition Language

DDL – Data Definition Language

- CREATE, ALTER, DROP, TRUNCATE
- CONSTRAINTS, DATA TYPES

DROP TABLE <TABLE NAME>;

Example:

DROP TABLE DEPT;

DDL – Data Definition Language

DDL – Data Definition Language

- CREATE, ALTER, DROP, TRUNCATE
- CONSTRAINTS, DATA TYPES

TRUNCATE TABLE <TABLE NAME>;

EXAMPLE:

TRUNCATE TABLE EMP;

DDL – Data Definition Language

DDL – Data Definition Language
- CREATE, ALTER, DROP, TRUNCATE
- CONSTRAINTS, DATA TYPES

Types of Constraints:
1. DEFAULT
2. NOT NULL
3. UNIQUE
4. PRIMARY KEY
5. FOREIGN KEY
6. CHECK

Example:

```
CREATE TABLE DEPT
(DEPTNO NUMBER(2) ,
DNAME VARCHAR2(14) NOT NULL,
LOC VARCHAR2(13),
CONSTRAINT PK_DEPT PRIMARY KEY (DEPTNO)
CONSTRAINT CHK_DEPT CHECK (LOC <> 'PYONGYANG');
```

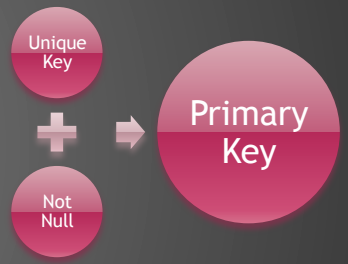
DDL – Data Definition Language

DDL – Data Definition Language

- CREATE, ALTER, DROP, TRUNCATE
- CONSTRAINTS, DATA TYPES

UNIQUE KEY Vs PRIMARY KEY

UNIQUE KEY + NOT NULL => PRIMARY KEY



Example:

```
CREATE TABLE DEPT
(DEPTNO NUMBER(2) ,
DNAME VARCHAR2(14) NOT NULL , -- UNIQUE also possible
LOC VARCHAR2(13),
CONSTRAINT PK_DEPT PRIMARY KEY (DEPTNO)
CONSTRAINT CHK_DEPT CHECK (LOC <> 'PYONGYANG');
```

DDL – Data Definition Language

DDL – Data Definition Language
- CREATE, ALTER, DROP, TRUNCATE
- CONSTRAINTS, DATA TYPES

Types of Data Types:
1. String Data Types – VARCHAR2, CHAR, CLOB
2. Number Data Type – NUMBER, INT, SMALLINT
3. DATE, TIMESTAMP, TIMESTAMP WITH TIMEZONE
4. RAW, BLOB
5. BFILE
6. ROWID

Note: Compete list is here –
https://docs.oracle.com/cd/B28359_01/server.111/b28286/sql_elements001.htm#SQLRF50950

Example:

```
CREATE TABLE DEMO
(DEMO_COL1 ROWID,
DEMO_COL2 <Data Type>
…
);
```

DML – Data Manipulation Language

- DML – Data Manipulation Language
  - INSERT, UPDATE, DELETE, MERGE

INSERT INTO <Table Name>
VALUE (COL1_VALUE, COL2_VALUE, COL3_VALUE….);

Example:
INSERT INTO DEPT VALUES (10, 'ACCOUNTING', 'NEW YORK');

DML – Data Manipulation Language

- DML – Data Manipulation Language
  - INSERT, UPDATE, DELETE, MERGE

UPDATE <Table Name>
SET <COL1> = <VALUE> | <EXPRESSION>;

Example:
UPDATE EMP
SET SAL = SAL * 1.10;

DML – Data Manipulation Language

DML – Data Manipulation Language

- INSERT, UPDATE, DELETE, MERGE

DELETE FROM <TABLE NAME>
WHERE <CONDITION>

Example:
DELETE FROM EMP
WHERE DEPTNO = 30;

DML – Data Manipulation Language

DML – Data Manipulation Language

- INSERT, UPDATE, DELETE, MERGE

```
MERGE INTO <TARGET Table Name>
USING <SOURCE TABLE>, <TARGET TABLE>
ON JOIN CONDITION
WHEN MATCHED THEN
        UPDATE SET COL1 = <VALUE> | <EXPRESSION>;
```

Example:

```
MERGE INTO DEPT_PARENT
USING DEPT_PARENT, DEPT_CHILD
ON JOIN CONDITION
WHEN MATCHED THEN
        UPDATE SET LOC = DEPT_PARENT.LOC || ',' || DEPT.CHILD.LOC;
```

DCL – Data Control Language

DCL – Data Control Language

- GRANT, REVOKE

SYNTAX:

GRANT <PRIVS> ON <OBJECT NAME>
TO <USER>

EXAMPLE:

GRANT SELECT, INSERT, UPDATE ON EMP
TO SCOTT;

DCL – Data Control Language

DCL – Data Control Language

- GRANT, REVOKE

SYNTAX:

```
REVOKE <PRIVS> ON <OBJECT NAME>
FROM <USER>
```

EXAMPLE:

```
REVOKE INSERT, UPDATE ON EMP
FROM SCOTT;
```

DQL – Data Query Langauge

DQL – Data Query Langauge

- SELECT, FETCH, DISTINCT,
- FILTER USING LIKE, IN, BETWEEN (Inclusive Range)
- GROUP BY, HAVING

SYNTAX:

```
SELECT <COL LIST>
FROM <TABLE NAME>
WHERE <CONDITION>
GROUP BY <COL NAME> -- used when grouping
HAVING <CONDITION> -- used along with group by
```

EXAMPLE:

```
SELECT e.DEPTNO, COUNT(*)
FROM  EMP e
GROUP BY e.DEPTNO;
```

DQL – Data Query Langauge

DQL – Data Query Langauge

- SELECT, FETCH, DISTINCT,
- FILTER USING LIKE, IN, BETWEEN (Inclusive Range)
- GROUP BY, HAVING

Note: REGEXP_LIKE function – regular expression introduced in Oracle 10g verion database.

EXAMPLE:

```
SELECT e.DEPTNO, COUNT(*)
FROM  EMP e
WHERE REGEXP_LIKE(ename, 'LL|TT','I');
```

DQL – Data Query Langauge

DQL – Data Query Langauge

- SELECT, FETCH, DISTINCT,
- FILTER USING LIKE, IN, BETWEEN (Inclusive Range)
- GROUP BY, HAVING

Note: BETWEEN has LOWER and UPPER bound and both are included in the search result.

EXAMPLE:

```
SELECT e.DEPTNO, COUNT(*)
FROM  EMP e
WHERE e.SAL BETWEEN 1500 AND 2850;
```

DQL – Data Query Langauge

Precedence and execution order

NOTE: SELECT gets you the columns, FROM gets you the rows

1. FROM
   - EMP E
2. WHERE
   - EMP.DEPTNO IN (10,20,30)
3. SELECT
   - EMP.DEPTNO, SUM(EMP.SAL)
4. GROUP BY
   - EMP.DEPTNO
5. HAVING
   - SUM(EMP.SAL) > 10000
6. ORDER BY
   - EMP.DEPTNO,SUM(EMP.SAL) OR 1,2

Operators in SQL

OPERATORS

- ARITHEMETIC - =, - , *, /, %
- COMPARISON - <, >, <=, >=, <>, !=
- LOGICAL – AND, OR, NOT, IN, BETWEEN, IS NULL, LIKE

SYNTAX:

```
SELECT <COL LIST>
FROM <TABLE NAME>
WHERE <CONDITION> -- OPERATORS
GROUP BY <COL NAME> -- used when grouping
HAVING <CONDITION> -- used along with group by
```

EXAMPLE:

```
SELECT e.DEPTNO, COUNT(*)
FROM  EMP e
GROUP BY e.DEPTNO;
```

JOINS

Joins are used when more than one table is required to fetch requirement information.

Types of Joins:

Equi-Join, Cross Join, Self-Join, Outer Join

SYNTAX:

```
SELECT <COL LIST>
FROM <TABLE NAME1>,<TABLE NAME2>
WHERE <JOIN CONDITION>
GROUP BY <COL NAME> -- used when grouping
HAVING <CONDITION> -- used along with group by
```

EXAMPLE:

```
SELECT e.EMPNO as  'EMPLOYEE_ID', e.ENAME as 'EMPLOYEE_NAME',
e.JOB as 'JOB', D.DEPTNO, D.DNAME as DEPT
  FROM  EMP e, DEPT d
WHERE  e.deptno = d.deptno;
```

JOINS

Joins are used when more than one table is required to fetch requirement information.

Types of Joins:

Equi-Join, Cross Join, Self-Join, Outer Join

SYNTAX:

```
SELECT <COL LIST>
FROM <TABLE NAME1>,<TABLE NAME2>;
```

EXAMPLE:

```
SELECT e.EMPNO as  'EMPLOYEE_ID', e.ENAME as
'EMPLOYEE_NAME', e.JOB as 'JOB', D.DEPTNO, D.DNAME as
DEPT
   FROM  EMP e, DEPT d;
```

JOINS

Joins are used when more than one table is required to fetch requirement information.

Types of Joins:

Equi-Join, Cross Join, Self-Join, Outer Join

SYNTAX:

```
SELECT <COL LIST>
FROM <TABLE NAME with alias>,<TABLE NAME1 with diff alias>
WHERE <JOIN CONDITION using alias>;
```

EXAMPLE:

```
SELECT e1.EMPNO as  'EMPLOYEE_ID', e1.ENAME as
'EMPLOYEE_NAME', e2.EMPNO as 'MGR NO', e2.ENAME as
'MANAGER_NAME'
  FROM  EMP e1, EMP e2
WHERE  e1.MGR = e2.EMPNO;
```

JOINS

Joins are used when more than one table is required to fetch requirement information.

Types of Joins:

Equi-Join, Cross Join, Self-Join, Outer Join

SYNTAX:

```
SELECT <COL LIST>
FROM <TABLE NAME with alias>,<TABLE NAME1 with diff alias>
WHERE <JOIN CONDITION using alias>;
```

EXAMPLE:

```
SELECT e1.EMPNO as  'EMPLOYEE_ID', e1.ENAME as
'EMPLOYEE_NAME', e2.EMPNO as 'MGR NO', e2.ENAME as
'MANAGER_NAME'
  FROM  EMP e1, EMP e2
WHERE  e1.MGR = e2.EMPNO(+); -- LEFT OUTER JOIN
```

TCL – Transaction Control Language

TCL – Transaction Control Language

- COMMIT, ROLLBACK, SAVEPOINT

SYNTAX:

```
COMMIT | ROLLBACK;

SAVEPOINT <SAVEPOINT NAME>
```

EXAMPLE:

```
SAVEPOINT A;
<SOME TRANSACTION>
SAVEPOINT B;
ROLLBACK TO SAVEPOINT A;
```

Functions

- STRING FUNCTIONS:
  - LENGTH, UPPER, LOWER, INITCAP, SUBSTR INSTR, CONCAT, RPAD, LPAD, LTRIM, RTRIM, TRIM, TRANSLATE, TO_CHAR
- DATE FUNCTIONS:
  - TO_DATE, ADD_MONTHS, MONTHS_BETWEEN
- GROUP/AGGREGATE FUNCTIONS:
  - COUNT, SUM, AVG, MIN, MAX
- ARITHEMATIC FUNCTIONS:
  - FLOOR, CEIL, ROUND, TRUNC

USAGE:

SELECT <Function_Name>(ARGUMENTS) FROM <TABLE NAME>;

VIEWS

- VIRTUAL TABLE WHICH CONTAINS ONLY METADATA.
- DATA IS FETCHED ON THE RUN TIME FROM THE BASE TABLES
- USED TO HIDE/CONCEAL SENSITIVE INFORMATION

EXAMPLE:

```
CREATE VIEW EMP_V
AS
SELECT EMPNO, ENAME,  JOB, DEPTNO FROM EMP
WHERE JOB NOT IN ('PRESIDENT', 'MANAGER');
```

TRIGGERS

- SET OF STATEMENTS WHICH EXECUTED (‘FIRED’) ON OCCURRENCE OF CERTAIN CONDITION OR EVENT

SYNTAX:

```
CREATE OR REPLACE TRIGGER <TRIGGER_NAME>
BEFORE | AFTER
INSERT | UPDATE | DELETE
ON <TABLE_NAME>
[FOR EACH ROW]
BEGIN
        <STATEMENTS>
END;
```

EXAMPLE:

```
CREATE TRIGGER SAL_RAISE
AFTER DELETE
ON DEPT
FOR EACH ROW
BEGIN
        UPDATE EMP SET DEPTNO = NULL WHERE DEPTNO = :OLD.DEPTNO;
END;
```

Quiz

TRUNCATE is a DDL command?

TRUE or FALSE?

What is the name STRING datatype?

What is the name STRING datatype?

VARCHAR2, CHAR

What are different types of triggers?

What are different types of triggers?

- BEFORE, AFTER
- INSERT , UPDATE, DELETE
- STATEMENT LEVEL, ROW LEVEL

What is a View?

A. A scenery

B. Same as Table

C. Database Objects

D. Virtual Table with only definition

E. A mirror image

What is a View?

A. A scenery

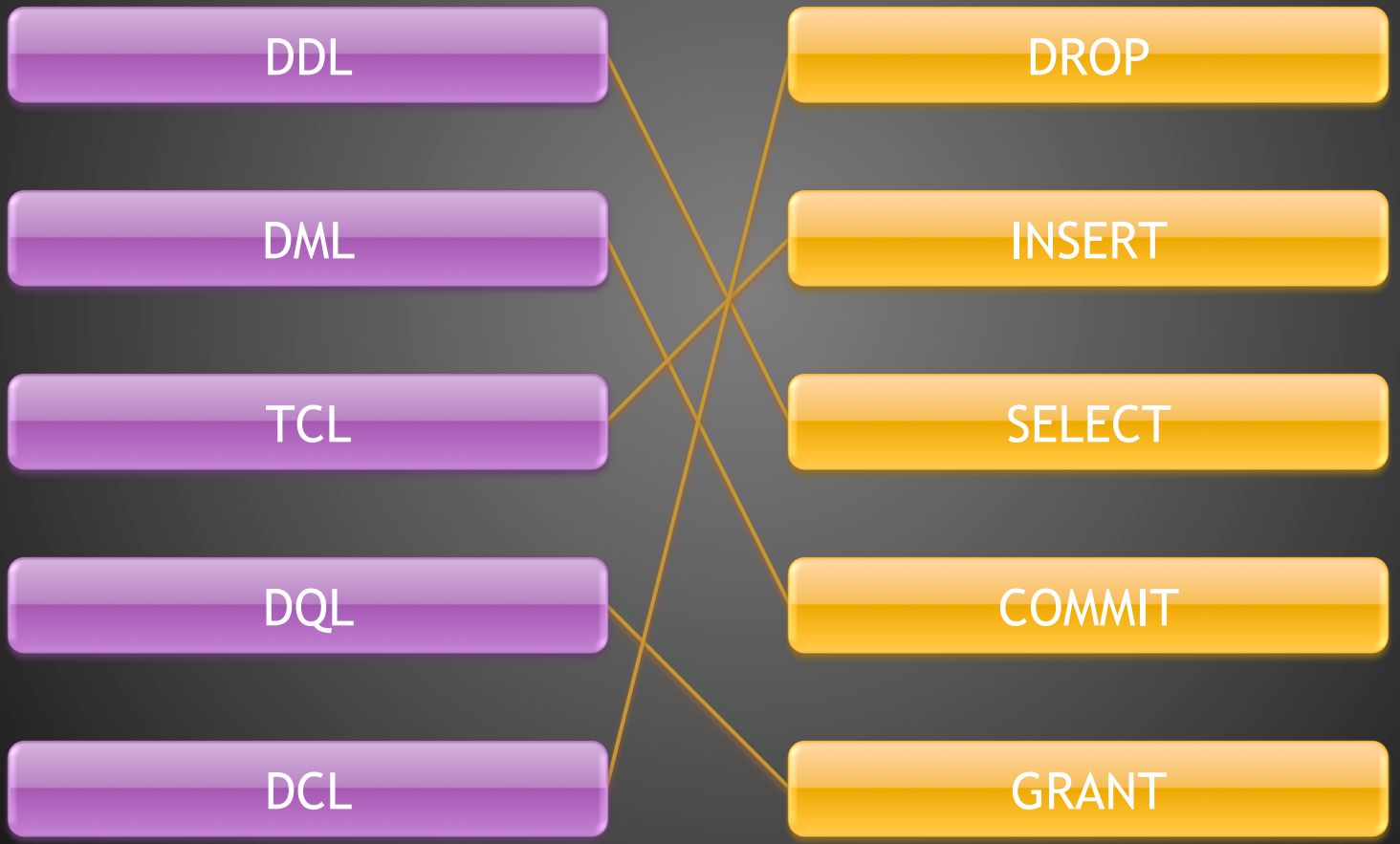
B. Same as Table

C. Database Objects

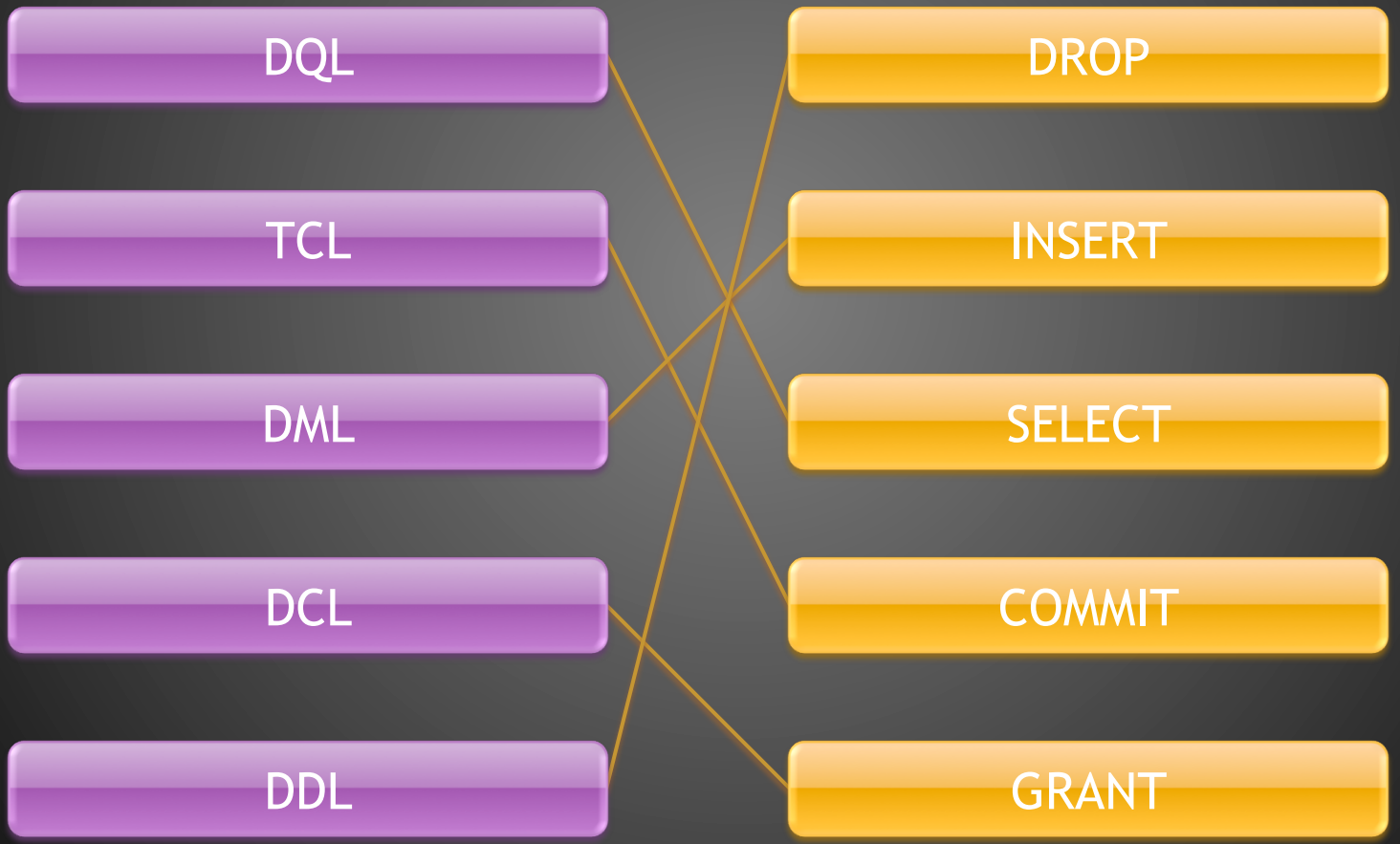
D. Virtual Table with only definition

E. A mirror image

Match the device to what it measures:



Match the device to what it measures:



What is a MUTATING TRIGGER?

A. A MUTE TRIGGER

B. TYPE OF TRIGGER

C. ERROR CONDITION IN TRIGGER

D. TRIGGER CONDITION AND EXECUTION STATEMENTS ARE CONFLICTING

E. A TRIGGER WHICH DOES NOT GET FIRED

What is a MUTATING TRIGGER?

A. A MUTE TRIGGER

B. TYPE OF TRIGGER

C. ERROR CONDITION IN TRIGGER

D. TRIGGER CONDITION AND EXECUTION STATEMENTS ARE CONFLICTING

E. A TRIGGER WHICH DOES NOT GET FIRED

What are Indexes?

What are Indexes?

Indexes are just like bookmarks, indexes and glossary in database which speeds up the fetching up of data.

Usually indexes are defined on Primary Key columns.

They can also be defined in multiple columns called which is called as composite indexes.

What are different types of Indexes?

What are different types of Indexes?

- Bitmap Indexes
- B-Tree Indexes
- Function Indexes
- Reverse key indexes

What is difference between functions and procedures?

What is difference between functions and procedures?

- Functions always return some value.
- Procedures may or may not return value.
- Functions can be called from SQL statements wherease procedures cannot be called.

Useful Links

Reference:
http://docs.oracle.com/

For Online SQL practise and demo:
http://apex.oracle.com/

For Quiz:
http://plsqlchallenge.oracle.com

For Understanding DBMS:
https://www.youtube.com/watch?v=FR4QIeZaPeM

Thank You

Wish you all the best!