Semestral project TDS I.

Oracle Academy – Dadabase Design & Programming with SQL

General project requirements:

1. Project in the scope of at least 10 tables
2. The entire project is written in English = names of tables, attributes, etc.
3. Tables contain all data types (text, numbers, dates, binary BLOB data)
4. The tables must be filled with a non-trivial number of data (a sufficient number to demonstrate all functionalities)
5. Create a database schema (logical and relational) in ORACLE SQL Developer Data Modeler http://www.oracle.com/technetwork/developer-tools/datamodeler/overview/index.html
6. Queries must be functional on the school's Oracle SQL server (dbsys.cs.vb.cz), to work with the database use only ORACLE SQL Developer http://www.oracle.com/technetwork/developer-tools/sql-developer/ overview/index-097090.html

The project must contain all the points listed below, of course more points can be solved in one task. You can comment on the project in English and additionally in Czech.

- DD S01 L02
  - Explain the difference between the concept of data and information - an example in your project written in English
- DD S02 L02
  - Entities, instances, attributes and identifiers - describe in examples on your project
- DD S03 L01
  - Describe all relations in your database in English, including cardinality and membership obligation - each relation in two sentences (page 10)
- DD S03 L02
  - Draw an ER diagram according to conventions
- DD S30 L04
  - Matrix diagram with relationships, draw for your solution (page 5)
- DD S04 L01
  - Supertypes and subtypes – define at least one instance of a supertype and a subtype in your project
- DD S04 L02
  - Description of business rules for your project
- DD S05 L01
  - Include at least one portable and one non-portable binding in your project
- DD S05 L03
  - Have at least one M:N relationship without information and one M:N relationship with information in your project
- DD S06 L01
  - Incorporate at least one 1:N identifying relationship into your project, with the fact that the transferred foreign key will also be the key in the new table
- DD S06 L02-04
  - Have your schema in first normal form - no non-atomic attributes
  - In second normal form – no subkey bindings
  - In third normal form - no links between secondary attributes
- DD S07 L01

    - Try to define ARC in your project (can be defined in ORACLE SQL Developer Data Modeler)
- DD S07 L02
    - Try to define hierarchical and recursive relations in your project
- DD S07 L03
    - Describe how you record historical data in your system
- DD S09 L01
    - Demonstrate saving changes over time on your project
- DD S09 L02
    - Try journaling in your project, i.e. saving past historical data (for example salary changes, workplace changes, etc.)
- DD S10 L01
    - Revise your design according to conventions for the readability of your schema
- DD S10 L02
    - Generic modeling – consider, possibly describe or use a generic model of data structures in your solution, how this approach is more advantageous compared to traditional data structure design methods
- DD S11 L01
    - Describe examples of integrity constraints on your project for entities, bindings, attributes, and user-defined integrity
- DD S11 L02-04
    - Generate a relational schema from your conceptual model and note the changes that have occurred in the schema and why
- DD S15 L01
    - Write query for concatenate strings by pipes || , and CONCAT()
    - SELECT DISTINCT
- DD S16 L02
    - WHERE condition for selecting rows
    - Functions LOWER, UPPER, INITCAP
- DD S16 L03
    - BETWEEN … AND
    - LIKE (%, _)
    - IN()
    - IS NULL, IS NOT NULL
- DD S17 L01
    - AND, OR, NOT
    - Evaluation priority ()
- DD S17 L02
    - ORDER BY atr [ASC/DESC]
    - Sorting by using one or more attributes
- DD S17 L03
    - Single row functions
    - Column functions MIN, MAX, AVG, SUM, COUNT
- SQL S01 L01
    - LOWER, UPPER, INITCAP
    - CONCAT, SUBSTR, LENGTH, INSTR, LPAD, RPAD, TRIM, REPLACE
    - Use virtual table DUAL
- SQL S01 L02
    - ROUND, TRUNC round for two decimal places, whole thousands MOD
- SQL S01 L03
    - MONTHS_BETWEEN, ADD_MONTHS, NEXT_DAY, LAST_DAY, ROUND, TRUNC

  - System constant SYSDATE
- SQL S02 L01
  - TO_CHAR, TO_NUMBER, TO_DATE
- SQL S02 L02
  - NVL, NVL2, NULLIF, COALESCE
- SQL S02 L03
  - DECODE, CASE, IF-THEN-ELSE
- SQL S03 L01
  - NATURAL JOIN, CROSS JOIN
- SQL S03 L02
  - JOIN … USING(atr)
  - JOIN .. ON (joining condition)
- SQL S03 L03
  - LEFT OUTER JOIN … ON ()
  - RIGHT OUTER JOIN … ON ()
  - FULL OUTER JOIN … ON ()
- SQL S03 L04
  - Joining 2x of the same table with renaming (link between superiors and subordinates in one table)
  - Hierarchical querying – tree structure of START WITH, CONNECT BY PRIOR, LEVEL dive
- SQL S04 L02
  - AVG, COUNT, MIN, MAX, SUM, VARIANCE, STDDEV
- SQL S04 L03
  - COUNT, COUNT(DISTINCT), NVL
  - Difference between COUNT (*) a COUNT (attribute)
  - Why using NVL for aggregation functions
- SQL S05 L01
  - GROUP BY
  - HAVING
- SQL S05 L02
  - ROLLUP, CUBE, ROUPING SETS
- SQL S05 L03
  - Multiple operations in SQL – UNION, UNION ALL, INTERSECT, MINUS
  - ORDER BY for set operations
- SQL S06 L01
  - Nested queries
  - Result as a single value
  - Multi-column subquery
  - EXISTS, NOT EXISTS
- SQL S06 L02
  - One-line subqueries
- SQL S06 L03
  - Multi-line subqueries IN, ANY, ALL
  - NULL values in subqueries
- SQL S06 L04
  - WITH .. AS() subquery construction
- SQL S07 L01
  - INSERT INTO Tab VALUES()
  - INSERT INTo Tab (atr, atr) VALUES()
  - INSERT INTO Tab AS SELECT …

- SQL S07 L02
  - UPDATE Tab SET atr= …. WHERE condition
  - DELETE FROM Tab WHERE atr=…
- SQL S07 L03
  - DEFAULT, MERGE, Multi-Table Inserts
- SQL S08 L01
  - Objects in databases – Tables, Indexes, Constraint, View, Sequence, Synonym
  - CREATE, ALTER, DROP, RENAME, TRUNCATE
  - CREATE TABLE (atr DAT TYPE, DEFAULT NOT NULL)
  - ORGANIZATION EXTERNAL, TYPE ORACLE_LOADER, DEFAULT DICTIONARY, ACCESS PARAMETERS, RECORDS DELIMITED BY NEWLINE, FIELDS, LOCATION
- SQL S08 L02
  - TIMESTAMP, TIMESTAMP WITH TIME ZONE, TIMESTAMP WITH LOCAL TIMEZONE
  - INTERVAL YEAT TO MONTH, INTERVAL DAY TO SECOND
  - CHAR, VARCHAR2, CLOB
  - about NUMBER
  - about BLOB
- SQL S08 L03
  - ALTER TABLE (ADD, MODIFY, DROP), DROP, RENAME
  - FLASHBACK TABLE Tab TO BEFORE DROP (view USER_RECYCLEBIN)
  - DELETE, TRUNCATE
  - COMMENT ON TABLE
  - SET UNUSED
- SQL S10 L01
  - CREATE TABLE (NOT NULL AND UNIQUE constraint)
  - CREATE TABLE Tab AS SELECT …
  - Own vs. system naming CONSTRAINT conditions
- SQL S10 L02
  - CONSTRAINT – NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY (atr REFERENCES Tab(atr)), CHECK
  - Foreign keys, ON DELETE, ON UPDATE, RESTRICT, CASCADE, etc.
- SQL S10 L03
  - about USER_CONSTRAINTS
- SQL S11 L01
  - CREATE VIEW
  - about FORCE, NOFORCE
  - WITCH CHECK OPTION
  - WITH READ ONLY
  - about Simple vs. Compex VIEW
- SQL S11 L03
  - INLINE VIEW Subquery in the form of a table SELECT atr FROM (SELECT * FROM Tab) alt_tab
- SQL S12 L01
  - CREATE SEQUENCE name INCREMENT BY n START WITH m, (NO)MAXVALUE, (NO)MINVALUE, (NO)CYCLE, (NO)CACHE
  - about ALTER SEQUENCE
- SQL S12 L02
  - CREATE INDEX, PRIMARY KEY, UNIQUE KEY, FOREIGN KEY
- SQL S13 L01
  - GRANT … ON … TO … PUBLIC
  - about REVOKE

  - What rights can be assigned to which objects? (ALTER, DELETE, EXECUTE, INDEX, INSERT, REFERENCES, SELECT, UPDATE) – (TABLE, VIEW, SEQUENCE, PROCEDURE)
- SQL S13 L03
  - Regular expressions
  - REGEXP_LIKE, REGEXP_REPLACE, REGEXP_INSTR, REGEXP_SUBSTR, REGEXP_COUNT
- SQL S14 L01
  - Transactions, COMMIT, ROLLBACK, SAVEPOINT
- SQL S15 L01
  - Alternative join notation without JOIN with join condition in WHERE
  - Left and right connection using atrA = atrB (+)
- SQL S16 L03
  - Recapitulation of commands and parameters - complete everything that was not mentioned in the previous points here