Xi'an Jiaotong-Liverpool University

西交利物浦大学

PAPER CODE	EXAMINER	DEPARTMENT	TEL
CPT103	Jianjun Chen	Computing	0512 81889137

2nd SEMESTER 2022/23 FINAL EXAMINATION

Undergraduate – Year 2

Introduction to Database Systems

TIME ALLOWED: 2 Hours

INSTRUCTIONS TO CANDIDATES

1、 This is a closed book examination.

2、 Total marks available are 100. This will count for 85% in the final assessment.

3、 Answer all questions.

4、 Answer should be written in the answer booklet(s) provided.

5、 Only English solutions are accepted.

6、 The university approved calculator - Casio FS82ES/83ES can be used.

7、 All materials must be returned to the exam supervisor upon completion of the exam. Failure to do so will be deemed academic misconduct and will be dealt with accordingly.

Question A: Basic SQL (30 marks)

Consider the following relations:

prizes

prize_id (int)	winner_sid (int)	award_time (date)
1	4121003	2020-01-12
2	6501396	2021-09-09
2	6501296	2021-09-09
3	6501296	2022-02-01

students

sid (int)	sname (varchar(20))	class_name (varchar(4))
4121003	Andrew	Y4C3
6501396	Henry	Y2C1
6501296	Bob	Y2C1
6501300	John	Y2C2
5501778	Jack	Y3C5

classes

class_name (varchar(4))	classroom (int)
Y2C1	201
Y2C2	202
Y2C3	203
Y3C4	304
Y3C5	305
Y4C3	403

a) You are given the following SELECT queries. What are the results of application of these queries to the tables "prizes", "classess" and "students"? Provide the answer in a table format. In case that query is not valid, explain the reason. (3 marks each)

1. `SELECT DISTINCT prize_id FROM prizes WHERE ID <> 3;`

There is no column prize_id called ID in Table prizes.

2. `SELECT sid FROM students WHERE class_name LIKE "Y_C1";`

sid
6501396
6501296

3. `SELECT sid FROM classes RIGHT OUTER JOIN students USING class_name;`

sid
4121003
6501396
6501296
6501300
5501778

4. `SELECT count(prize_id) FROM prizes NATURAL JOIN students GROUP BY class_name;`

count(prize_id)
1
3

b) Write an SQL query to get the number of prizes won by each student. In the result, list students' ID numbers as well as the number of prices won. (4 marks)

SELECT S.sid, count(p.prize_id) from students S LEFT JOIN prizes P on S.Sid = P.winner_sid Group BY S.Sid.

c) Write an SQL query to get all students called "Bob" who are currently in year 2. In the result, list student IDs. (4 marks)

SELECT sid from students where sname = 'Bob' AND class_Name like '%Y2%';

d) Write an SQL query to list classes that have not won any prizes after 2020-12-09. In the result, list the class_name in the descending order. (5 marks)

SELECT C.class_name from classes c left outer join students S on c.class_name = S.class_name where S.Sid NOT IN (SELECT winner_sid from prizes where award_time > '2020-12-09') ORDER BY C.class_name DESC;

e) Write an SQL query to list classes along with their neighbouring classrooms. A classroom is considered to be the neighbouring classrooms of another if the difference between their room numbers is 1. In the result, list class names and the two neighbouring classrooms. Non-existing neighbouring classrooms should be listed as NULL. The supposed output of your query is given below using the data in the current classes table: (5 marks)

class_name	neighbour1	neighbour2
Y2C1	NULL	202
Y2C2	201	203
Y2C3	202	NULL
Y3C4	NULL	305
Y3C5	304	NULL
Y4C3	NULL	NULL

SELECT S1.class_name, S2.classroom as neighbour1, S3.classroom as neighbour2 from classes S1 left outer join classes S2 on S1.classroom - 1 = S2.classroom left outer join classes S3 on S1.classroom + 1 = S3.classroom ORDER BY S1.classroom;

SELECT C1.class_name FROM classes C1 where C1.class_name NOT IN (SELECT S.class_name from students S Join Prizes P on S.Sid=P.winner_sid WHERE P.award_time > '2020-12-09') ORDER BY C1.class_name DESC;

Question B: Transactions and Recovery (20 marks)

Answer the following questions related to transactions and recovery:

a) What does "COMMIT" mean in transactions? (6 marks)
b) Why supporting concurrency is important for database? (4 marks)
c) Describe the "lost update" problem of concurrency using an example. (10 marks)

Question C: Normalisation (20 marks)

Normalise the following table "T" into the 3rd Normal Form by clearly describing the normalisation process, i.e. the dependencies removed and how the table is split into sub-tables. Describe the primary key and functional dependencies for each resulting sub-tables.

A	B	C	D	E	F	G

Attributes (A, B, C) form the primary key and the functional dependencies:

```
A, B, C -> D, E, F, G

B, C -> E, F

F -> C

D -> G
```

2NF: FD B,C → E,F is a partial dependence, after removing it, table T becomes

T_1: (A, B, C, D, G) with primary key (A, B, C)

and T_2: (B, C, E, F) with primary key (B, C)

3NF: column G is transitively depend on primary key via D in T_1, after removing FD D → G.

T_1 becomes $T_{1\text{-}1}$ (A, B, C, D) with primary key (A, B, C)

and $T_{1\text{-}2}$ (D, G) with primary key (D).

Final design:

$T_{1\text{-}1}$ (A, B, C, D) with primary key (A, B, C) T_2 (B, C, E, F) with primary key (B, C)

$T_{1\text{-}2}$ (D, G) with primary key (D)

Question D: Entity-relationship Modelling (30 marks)

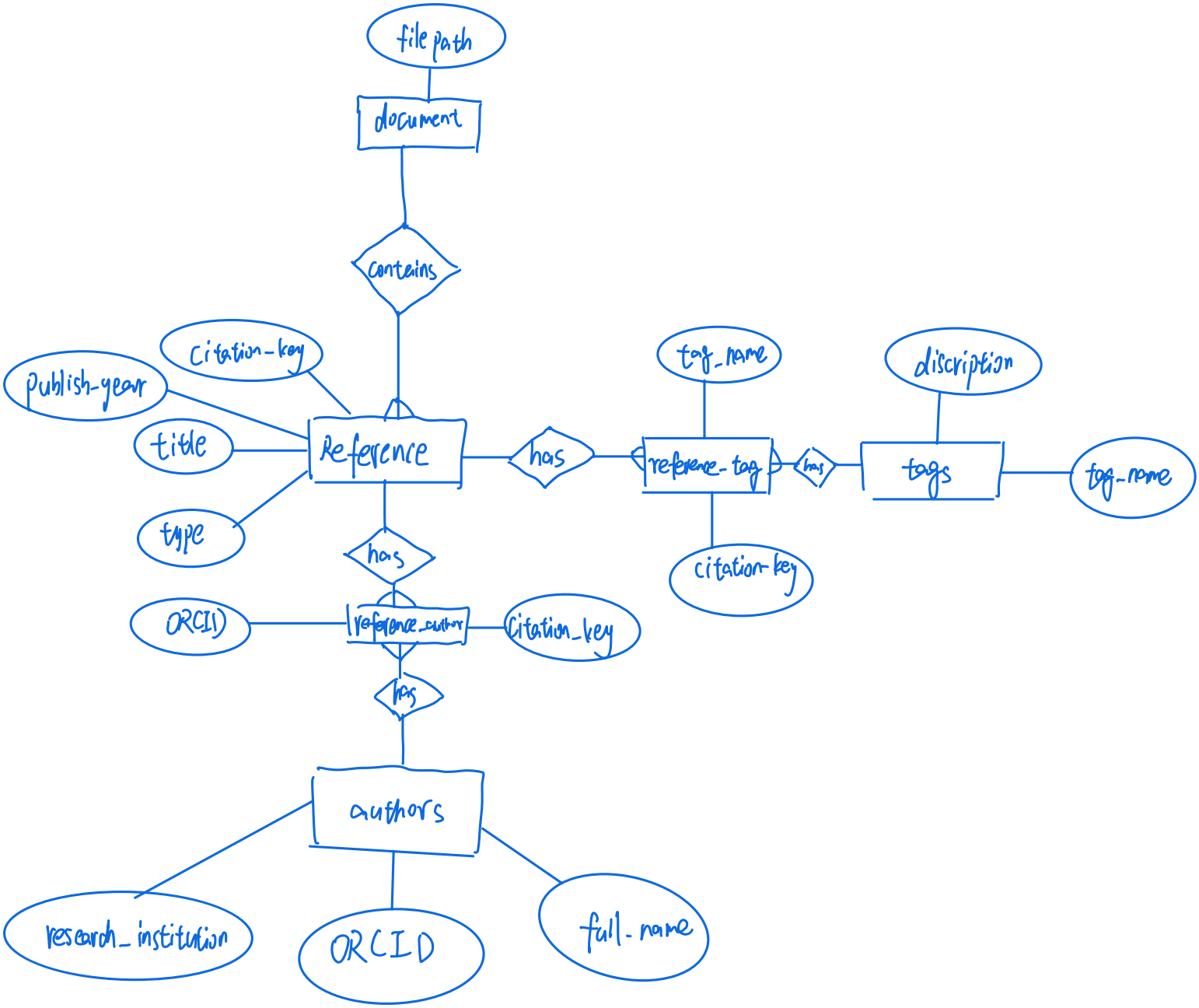
You are asked to develop a database that is used by a reference manager application. Reference manager applications are tools used to create, organize and store references for conference papers or journal articles. The requirements for the database are provided below:

1. The whole reference library is managed using tags. Tags are up to 15 characters long.
    1. Each reference can have zero or more tags.
    2. A same tag can be added to one or more references.
    3. Each tag can be described with some comments.
2. Each reference has a unique citation key, a list of authors, a publish year, a title, the source of this reference (the name of the journal or conference), the type of reference (journal article or conference paper) and optionally a file path if the user owns the document.
3. Each author has a unique ID called ORCID, a full name and the research institution he/she is currently working in.
4. ORCID is always 16 digits long, the last digit can be the letter 'x'.
5. Sometimes a document contains a collection of articles.

Task 1: Draw the entity relationship diagram. (20 marks)

Task 2: Based on your solution to Task 1 above, write the SQL code to create the tables for the database. You should include all the specified attributes and specify the appropriate primary and foreign keys. All primary keys and foreign keys must be added separately using ALTER statements. Minor syntactical errors in your SQL code will not be penalised in the marking of this answer. (10 marks)

END OF FINAL EXAM


file path
document
contains
citation-key
publish-year
title
type
Reference
has
tag_name
discription
reference_tag
has
tags
tag-name
citation-key
has
ORCID
reference_author
citation_key
has
authors
research_institution
ORCID
full_name

```
CREATE TABLE authors(
    ORCID CHAR(16) NOT NULL,
    full-name VARCHAR(20) NOT NULL,
    research_insitution VARCHAR(200)
    );

CREATE TABLE document (
    file_path VARCHAR(200) NOT NULL
    );

CREATE TABLE reference-author (
    ORCID CHAR(16) NOT Null,
    Citation_key VARCHAR(200) NOT NULL
);

CREATE TABLE reference-tag (
    tag_name CHAR(15) NOT NULL,
    Citation_key VARCHAR(200) NOT NULL
    );

CREATE TABLE tags (
    discription VARCHAR(200),
    tag_name CHAR(15) NOT NULL
    );

CREATE TABLE Reference (
    Citation_key VARCHAR(200) NOT NULL,
    publish_year DATE NOT NULL,
    title VARCHAR(60) NOT NULL,
    type VARCHAR(30) NOT NULL,
    file_path VARCHAR(200)
    Constraint check_type CHECK (type='journal article'
                                 OR type='conference paper');
    );

ALTER TABLE authors ADD CONSTRAINT ORCID primary KEY.
ALTER TABLE Reference ADD CONSTRAINT Citation_key primary KEY.
ALTER TABLE document ADD CONSTRAINT file_path primary KEY.
ALTER TABLE tags ADD CONSTRAINT tag_name primary KEY.

ALTER TABLE reference-author ADD CONSTRAINT PK1 foreign KEY (Citation_key) references Reference (Citation_key);
ALTER TABLE reference-author ADD CONSTRAINT PK2 foreign KEY (ORCID) references authors (ORCID);
ALTER TABLE reference-tag ADD CONSTRAINT PK3 foreign KEY (tag_name) references tags (tag_name);
ALTER TABLE reference-tag ADD CONSTRAINT PK4 foreign KEY (Citation_key) references Reference (citation_key)
ALTER TABLE Reference ADD CONSTRAINT PK5 foreign KEY (file_path) references document (file_path);
```