
Database Systems: Ph.D. Written Qualifying Exam.

Jan. 2014

---

1. (20 points) Database Concepts (choose one):

   1.1 Explain the three-layer database architecture.

   1.2 Discuss the six relational integrity constraints.

2. (20 points) Database Design:

   Show how the below company data are modeled into (1) an ER schema and (2) create relational tables in DDL/SQL.

   'In a university, a student belongs to a department and may participate in up to two research projects based on different work_hours (such as half_time, full_time, etc.). Projects are conducted by a department, and a project must have a minimum of four students.' (You may design the entities' attributes by yourself.)

3. (20 points) Query Processing:

   From the database above, a database application issues a query below:

   "Find female students who work for the CS department, and give their SIDs, names, and locations of their department."

   Show (1) the query in DML/SQL, and (2) query plan in the form of a query tree (or step-wise algebra operations).

4. (20 points) Transaction Management (choose one):

   4.1 Transaction Recovery: (1) interpret the transaction log below onto a time chart; (2) show transaction recovery process for Deferred Update algorithm.

TRANSACTION LOG:

Log Id.	Tr. No.	Prv ptr	Next ptr	Operation	Table	Row id.	Attr.	Before val.	After val.
…	…	…	…	□□□□	….	…	…	…	…
220	79	Null	228	START	-	-	-	-	-
221	77	119	249	INSERT	COURSE	606	-	-	606,DB, …
228	79	220	230	INSERT	STUDENT	20072678	-	-	20072678, KIM, F, …
230	79	228	Null	COMMIT	-	-	-	-	-
231	89	Null	240	START	-	-	-	-	-
240	89	231	262	UPDATE	STUDENT	20071398	GPA	3.0	3.5
249	77	221	260	DELETE	COURSE	600	-	600,OS, …	-
255	checkpoint								
260	77	249	266	UPDATE	COURSE	606	ROOM	L305	L405
262	89	240	Null	COMMIT	-	-	-	-	-
266	77	260		UPDATE	COURSE	500	ROOM	L405	L305
!! Crash !!									

4.2 Transaction Concurrency: (1) Explain transaction serializability and (2) show a serializable schedule for transaction <Ti> using Two-phase Locking Protocol.

<Ti>

```
BEGIN_TR
request a read-lock on (A)
   read_lock (A) acquired;
   read (A)
   unlock (A)
request a write-lock on (B)
   write_lock (B) acquired;
   read_ (B);
   B:=A * B;
   write (B)
   unlock (B);
(COMMIT)
```

(Hint) Two-phase Locking Protocol: All locking operations (read_lock, write_lock) precede the first unlock operation in the transaction. After releasing a lock, a transaction must never go on to acquire any more locks. (growing/shrinking phase)

5. (20 points) Short-answer questions. (<u>choose five questions only</u>)

(a) Base data, auxiliary data, metadata
(b) View relation
(c) FD
(d) Normalization
(e) Primitive algebra operations
(f) JOIN operation
(g) Database transaction
(h) Schema