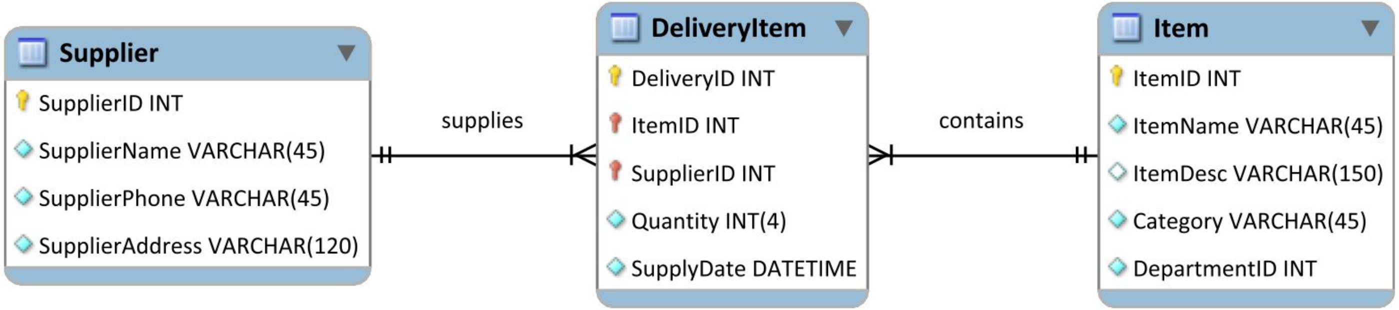
Database Systems

Tutorial Week 10

Objectives

I. Apply capacity planning concepts
II. Review of backup & recovery concepts
III. Apply backup & recovery concepts to case studies
IV. Review transactions concepts
V. Apply transactions concepts

Exercise — Capacity Planning (5 mins)



Consider the case of a department store. An analyst has determined that there are 50 distinct suppliers that provide 2000 distinct items to the store. They have determined that the average delivery is of 40 distinct items and that each supplier delivers approximately once a week (the analyst has estimated this to be 50 deliveries a year). For each delivery by a supplier, there are on average 40 rows added to the DeliveryItem table. While the Item and Supplier tables stay constant in size, the DeliveryItem table grows by 100,000 rows every year.

This assumes that suppliers and items stay constant; however, if the business is successful, the suppliers and frequency of deliveries and number of distinct items delivered can be expected to grow. If we know the length of each row, we can estimate how big each table will be year by year.

Exercise — Capacity Planning (5 mins)

Using information about data type storage from the MySQL documentation and information from the data dictionary, the analyst has determined the following average row lengths of each table:

Table	**Number of rows**	**Average row length**
Supplier	50 rows	144 bytes
Item	2000 rows	170 bytes
DeliveryItem	0 rows	19 bytes

Table 1: Row volume and row length for the Supplier delivers Item entities.

Assume that the number of suppliers and number of items do not change from year to year, and that the delivery schedule remains the same. Calculate the size of the three tables:

a. When database use begins (year 0)
b. After one year of database use
c. After five years of database use

Bytes → kilobytes, ÷ 1024
Kilobytes → megabytes, ÷ 1024

1 kb = 1024 bytes
1 mb = 1024 kb

Exercise — Capacity Planning

a. At the start
   i. Supplier
      = 50 rows × 144 bytes
      = 7,200 bytes
      ≈ 7 kb
   ii. Item
      = 2,000 rows × 170 bytes
      = 340,000 bytes
      ≈ 332 kb
   iii. DeliveryItem
      = 0 rows × 19 bytes
      = 0 bytes

Exercise — Capacity Planning

b. After 1 year

i. Supplier and Item tables stay constant
ii. DeliveryItem table grows by 100,000 rows a year
   1 year × 100,000 rows × 19 bytes
   = 1,900,000 bytes / 1024 / 1024
   ≈ 1.8 mb

Exercise — Capacity Planning

c. After 5 years

i. Supplier and Item tables stay constant
ii. DeliveryItem table grows by 100,000 rows a year
    5 years × 100,000 rows × 19 bytes
    = 9,500,000 bytes / 1024 / 1024
    ≈ 9.1 mb

Exercise — Capacity Planning

- Tables like DeliveryItem are “event tables” because they record events
- Examples of event tables include sales, deliveries, orders etc.
- We need to pay special attention to event tables when applying capacity planning concepts, because other tables aren’t expected to grow rapidly

Backup and Recovery Concepts

- Why do we need a backup of our database?
  - So we don’t lose data since data might be corrupt or lost

Backup and Recovery Concepts

- Backup type
  - Logical backup
    - Keeps a record of the data and also the *metadata* (column name, data types, type of indexes)
    - Takes more information than physical backups because they need to include the structure of the table
      - Its relationships to other entities
      - Index information
      - Data types
    - Useful when we want to move data from one operating system to another
      - Physical databases can’t do so since the file format is usually unique to each operating system
    - Also, the different database engines (MySQL, Oracle, IBM, Microsoft) aren’t physically compatible
      - Logical backups are very good for migrating data from one database to a completely different database and environment

Backup and Recovery Concepts

- Backup type
  - Physical backup
    - A direct copy of the physical files (copy paste)
    - Fastest way to make a copy of the database
    - A database administrator makes physical copies of the files and then stores this copy on a backup server or other media storage (SSD or HHD)
    - In the event of a database failure, the physical copies can be restored to their original location, and then transactions can be replayed using the Crash Recovery log
      - Crash Recovery log is an area in memory that records every change we make in the database
      - This log is continually written to disk to avoid any loss of information

Backup and Recovery Concepts

- Backup mode
  - Online backup
    - Users can still use the database and are unaffected by the backup operations
    - No loss of availability of the database
  - Offline backup
    - Database server process is shut down while the copy of the file is made
    - No users can connect to the database or process any queries while a database is offline

Backup and Recovery Concepts

- Backup location
  - Onsite backup
    - Backups are stored on the same premises, but on a different device as the database
  - Offsite backup
    - Backups are stored in a remote location
      - Usually more than 160 km away from the primary site

Backup and Recovery Concepts

- Full backups
  - Backup all data in the database
- Incremental backups
  - Only back up the changes made since the last full backup
  - Smaller in size and shorter in duration
  - Help for both recovery and performance of the database
- Partial backup
  - Backup only certain tables or schemas instead of backing up the entire database

Backup and Recovery Concepts

How does database recovery work?

Two phases:

1. Restore the backup to the database server machine
2. Recover until the point of the database failure using the recovery log
   (No committed data in the database should be lost)

Backup and Recovery Case Study (7 mins)

ACME Manufacturing makes widgets in its factory. The factory runs 24 hours a day, 7 days a week in three shifts. The quietest shift is the Sunday night shift, which runs from midnight Sunday to 8am Monday. While ACME manufactures widgets, the database must run. This is ACME's only widget factory.

The database administrator has implemented a backup policy that takes a full backup every Sunday at 3am in the nightshift and then an incremental backup on Tuesday, Thursday and Saturday mornings at 3am.

The backup strategy has determined that if there is a database failure, restoration of the database is time critical. ACME must have the shortest outage time to restore and recover the database. This means that the database must be restored quickly so that the manufacturing can continue. ACME must have the smallest elapsed time from the point of failure to the database being fully operational and useable.

a. Given the business requirements and the database administrator's backup policy, what type, mode and site would you recommend for ACME's database backups?

b. If the database suffered a media failure on Friday at 9:23am, how many backups would need to be restored?

c. Given the same failure, what would be the benefits and costs of changing the backup strategy to do full backups on Sunday, Tuesday, Thursday and Saturday mornings at 3am?

Backup and Recovery Case Study (7 mins)

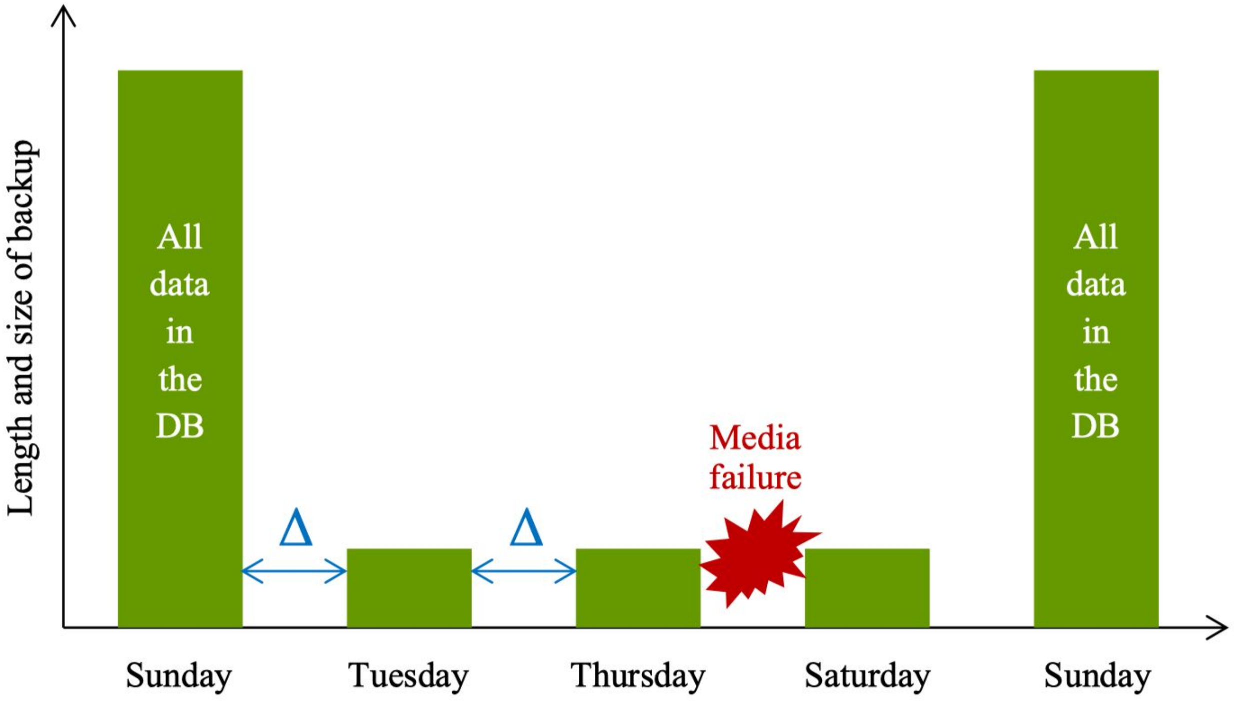


Figure 1. A timeline of full and incremental backups showing the media failure on Friday morning.

Backup and Recovery Case Study (7 mins)

a. Given the business requirements and the database administrator’s backup policy, what database backup type, mode and site would you recommend?

- Mode
  - Online
  - No interruption of operations at ACME
- Location
  - Onsite
  - This is ACME’s only widget factory
- Physical or logical
  - Physical
  - Fastest to backup and restore

Backup and Recovery Case Study (7 mins)

b. Consider the Full and Incremental backup timeline in Figure 1. If the database suffered a media failure on Friday at 9:23am, how many backups would need to be restored?

- 3 backups:
  - The Sunday full backup
  - Tuesday incremental backup
  - Thursday incremental backup
- Then replay the crash recovery logs from Thursday morning until Friday 9.23am

Backup and Recovery Case Study (7 mins)

c. Given the same failure, what would be the benefits and costs of changing the backup strategy to do full backups on Sunday, Tuesday, Thursday and Saturday mornings at 3am?

- Pros:
    - Faster restoration process since it would reduce the time to fully restore the database from the last full backup and apply the crash recovery logs
- Cons:
    - More space would be required to backup the database
    - The backup will also take longer to complete since there’s more data to copy
    - If we don’t remove backups from the database server, we could fill up the file system

Transactions Concepts

- What’s a transaction?
    - Set of SQL statements (“logical unit of work”) that must be fully completed or aborted
    - Usually corresponds to a single “action” that involves several changes to the database
        - E.g. removing funds from one account and adding them to another
    - Adheres to ACID principles

Transactions Concepts

- ACID
  - Specifies four desirable properties of transactions
  - A DBMS that implements transactions that achieve these properties is “ACID-compliant”
  - Atomic
    - Each transaction entirely succeeds or entirely fails
    - If a failure occurs mid-transaction, the DBMS must discard (roll back) all of the transaction’s changes up to that point
  - Consistent
    - Upon completion, a transaction must respect data integrity rules (constraints) and leave the database in a consistent state
  - Isolated
    - Changes made in one transaction can’t be seen from within other transactions
    - Transactions give the impression of being executed side-by-side simultaneously, but in practice, data locking means some transaction might be delayed by other transactions as they wait for locks to be released
  - Durable
    - Once a transaction is committed, the inserts, updates and deletes carried out in that transaction persist permanently in the database

Transactions Concepts

- Concurrency
  - The ability to allow many users to connect to and work with the database simultaneously
  - Possible for a database to allow concurrent access and still satisfy the ACID properties
    - But the isolation property is tricky

Transactions Concepts

- The lost update problem
  - When two users try to update the same piece of data in the database at the same time, they might conflict with each other
  - One user changes the data value, and the other user doesn't see this update before writing their own update to that database
  - The first user's update is lost

Transactions Concepts

- The lost update problem
  - E.g. two people are accessing the same bank account
  - Suppose my bank account contains $500
  - I'm at a store buying a $300 TV
  - At the same time, my employer is paying me $120 in wages

My transaction at the electronics store	**My employer's transaction**
Read account balance ($500)	
	Read account balance ($500)
Write account balance less $300 ($200)	
	Write account balance plus $120 ($620)

  - I now have $620 in my account!

Exercise — Transactions (15 mins)

It's class registration day, when UniMelb students register in tutorial classes for the upcoming semester. In one particular subject, each tutorial class can fit a maximum of 24 students.

Eamonn and Jacqueline both wish to register in the Wednesday 10am tutorial class for this subject. This class already has 23 students enrolled – just one place remains.

Suppose the database contains tables like this:

FK
TutorialClass (<u>SubjectCode</u>, <u>TutorialNumber</u>, TotalEnrolments)

FK FK FK
TutorialEnrolment (<u>SubjectCode</u>, <u>TutorialNumber</u>, <u>StudentNumber</u>)

a. Describe how a lost update could occur in this database when Eamonn and Jacqueline try to simultaneously register in the Wednesday 10am tutorial.
b. How could the lost update problem be avoided in this situation?

Exercise — Transactions (15 mins)

a. Describe how a lost update could occur in this database when Eamonn and Jacqueline try to simultaneously register in the Wednesday 10am tutorial.

- Suppose Eamonn's enrolment request is received a split second before Jacqueline's
- The server might execute the operations in this order:

Eamonn	**Jacqueline**
Read TutorialClass.TotalEnrolments (23)	
	Read TutorialClass.TotalEnrolments (23)
Insert row into TutorialEnrolment	
	Insert row into TutorialEnrolment
	Write TutorialClass.TotalEnrolments (24)
Write TutorialClass.TotalEnrolments (24)	

- Even though there are now 25 students enrolled in the class, the value of TotalEnrolments for the class is 24
- A lost update has occurred

Exercise — Transactions (15 mins)

b. How could the lost update problem be avoided in this situation?

- Could enforce serial execution, where only one transaction is executed at a time
  - This would make the system very inefficient, but it would guarantee that the isolation property of ACID is satisfied
- A better solution is to use locking
  - When a transaction wants to read the TotalEnrolments value for a class, it takes out a lock on that *row* of the TutorialClass table
  - This prevents other transactions from modifying that row
  - Once the transaction has finished writing to the row, it releases the lock
  - This approach is more efficient than serial execution
    - A student's enrolment request doesn't have to wait for all other requests in the system
    - It only needs to wait for the completion of other requests to enrol in the same class

Exercise — Transactions (15 mins)

- Could use other concurrency control methods mentioned in the lecture
    - Timestamps
        - Assign a unique timestamp to each transaction
        - If the timestamp of TotalEnrolments changes between when Eamonn reads it and when he is about to write it, Eamonn's transaction would abort and restart
    - Optimistic concurrency control
        - If TotalEnrolments is no longer equal to its original value when Eamonn is about to write it, Eamonn's transaction would abort and restart

Week 10 Lab

- Canvas → Modules → Week 10 → Lab → L10 Transactions (PDF)
- Objectives:
  - Practice SQL transactions using the MySQL Workbench Query Browser
  - Run multiple MySQL Workbench connections to one database to demonstrate concurrency
- Breakout rooms, “ask for help” button if you need help or have any questions