16 Databases and SQL

21 Databases

1. Why Use Databases
   - databases store large datasets efficiently
   - they provide fast, reliable data access even for massive tables
   - using databases avoids loading entire datasets into r at once
2. Connecting To A Database
   - dbplyr works with dplyr syntax but translates it to sql
   - src_dbi() or DBI::dbConnect() creates a connection object
   - tbl() references a table without pulling data into memory
3. Lazy Evaluation
   - dbplyr never runs operations immediately
   - transformations create a lazy query that is only executed on collect()
   - this avoids unnecessary computation and improves performance
4. dplyr Verbs Become SQL
   - filter(), select(), mutate(), arrange(), and summarize() translate directly to sql
   - dbplyr automatically generates efficient sql under the hood
   - show_query() displays the sql that will be executed
5. Collecting Data
   - collect() pulls the results of a sql query into r
   - only use collect() when the dataset is small enough for memory
   - until then, operations remain remote and database-backed
6. Writing Data To Databases
   - copy_to() uploads a local dataframe to a database
   - dbWriteTable() writes a table using dbi
   - careful naming and indexing help optimize storage and queries
7. SQL Limitations And Workflows
   - sql may not support every dplyr function
   - some operations must be rewritten to fit sql's capabilities
   - keeping computations inside the database is more efficient
8. Best Practices For Using Databases With R
   - always inspect schema and available tables before querying
   - limit data early using select() and filter()
   - use indexes on database columns for faster queries
   - validate that generated sql behaves as expected using show_query()