Databases

A-Level Computer Science · Vocabulary in context

Word	Example
flat files /flæt faɪlz/	**Flat files** store all the data in one table.
data redundancy /ˈdeɪtə rɪˈdʌndənsi/	**Data redundancy** means the same data is stored twice.
data inconsistency /ˈdeɪtə ˌɪnkənˈsɪstənsi/	**Data inconsistency** means two copies of the data do not match.
field /fiːld/	The **field** “Name” stores each student’s name. ”Name”
integrity /ɪnˈtegrɪti/	Data **integrity** means the data is correct and complete.
relational database /rɪˈleɪʃənl ˈdeɪtəbeɪs/	A **relational database** has linked tables.
table /ˈteɪbl/	The STUDENT **table** has one row for each student. STUDENT
DBMS /ˌdiː biː em ˈes/	A **DBMS** is software that manages a database.
query /ˈkwɪərɪ/	A **query** finds all students in class 10A. 10A
SQL /ˌes kjuː ˈel/	**SQL** is a language for asking a database questions.
entity /ˈentɪti/	A student is an **entity** in the school database.
record /ˈrekɔːd/	Each **record** in the school database holds one student’s details.
tuple /ˈtuːpl/	A **tuple** is one row of a table.

exercises.lol exercises.com.cn

Word	Example
attribute /ˈætrɪbjuːt/	Name and age are each an **attribute** of a student.
primary key /ˈpraɪməri kiː/	StudentID is the **primary key**: no two students share it. StudentID StudentID。
foreign key /ˈfɒrən kiː/	ClassID in the STUDENT table is a **foreign key**. STUDENT ClassID
composite key /ˈkɒmpəzɪt kiː/	A **composite key** is made of two fields together.
candidate key /ˈkændɪdeɪt kiː/	A field that could be the primary key is a **candidate key**.
secondary key /ˈsekəndəri kiː/	Surname is a **secondary key** used for searching. Surname
indexing /ˈɪndeksɪŋ/	**Indexing** makes searches faster.
join /dʒɔɪn/	A **join** puts data from two tables together.
referential integrity /ˌrefəˈrenʃl ɪnˈtegrɪti/	**Referential integrity** stops a foreign key pointing to nothing.
entity-relationship diagram /ˈentɪti rɪˈleɪʃənʃɪp ˈdaɪəgræm/	An **entity-relationship diagram** shows how the tables link.
cardinality /ˌkɑːdɪˈnælɪti/	The **cardinality** is one-to-many: one class has many students.
one-to-many /wʌn tə ˈmeni/	One teacher has many students: a **one-to-many** relationship.
link table /lɪŋk ˈteɪbl/	A **link table** breaks a many-to-many relationship into two.
normalisation /ˌnɔːməlaɪˈzeɪʃn/	**Normalisation** removes repeated data from tables.

Word	Example
normal forms /ˈnɔːml fɔːmz/	1NF, 2NF and 3NF are **normal forms**. 1NF、2NF 3NF
atomic /əˈtɒmɪk/	An **atomic** field holds only one value.
transitive dependency /ˈtrænsɪtɪv dɪˈpendənsi/	In a **transitive dependency**, a field depends on another non-key field.
partial dependency /ˈpɑːʃl dɪˈpendənsi/	In a **partial dependency**, a field depends on only part of the key.
data dictionary /ˈdeɪtə ˈdɪkʃənəri/	The **data dictionary** lists every field and its type.
concurrent access /kənˈkʌrənt ˈækses/	**Concurrent access** lets many users use the database at once.
transactions /trænˈsækʃnz/	The bank's **transactions** must not be lost.
backup /ˈbækʌp/	Make a **backup** of your work every day.
views /vjuːz/	**Views** show each user only the data they need.
data management /ˈdeɪtə ˈmænɪdʒmənt/	The DBMS does **data management** for us. DBMS
data modelling /ˈdeɪtə ˈmɒdəlɪŋ/	**Data modelling** plans the tables before we build them.
logical schema /ˈlɒdʒɪkl ˈskiːmə/	The **logical schema** shows the tables and their links.
data integrity /ˈdeɪtə ɪnˈtegrɪti/	**Data integrity** means the data is correct.
data security /ˈdeɪtə sɪˈkjʊərɪti/	**Data security** keeps the data safe from attackers.

exercises.lol exercises.com.cn

Word	Example
query processor /ˈkwɪərɪ ˈprəʊsesə/	The **query processor** runs your SQL. SQL
developer interface /dɪˈveləpə ˈɪntəfeɪs/	The **developer interface** helps you build tables and forms.
authentication /ɔːˌθentɪˈkeɪʃn/	A password is a kind of **authentication**.
Data Definition Language /ˈdeɪtə ˌdefɪˈnɪʃn ˈlæŋgwɪdʒ/	CREATE TABLE is part of the **Data Definition Language**. CREATE TABLE
Data Manipulation Language /ˈdeɪtə məˌnɪpjʊˈleɪʃn ˈlæŋgwɪdʒ/	SELECT and UPDATE are part of the **Data Manipulation Language**. SELECT UPDATE
aggregate functions /ˈægrɪgeɪt ˈfʌŋkʃnz/	SUM and COUNT are **aggregate functions**. SUM COUNT