**Data Warehousing**

1. **Relational Databases for Operational Processing用于操作处理的关系数据库**
2. **Used to run day to day business operations**
3. **Automation of routine business processes**

* **Accounting**
* **Inventory**
* **Purchasing**
* **Sales**

1. **Created huge efficiencies**
2. **OLTP Databases**
3. **OLTP = “OnLine Transaction Processing”** **在线交易处理**
4. **Transaction processing supports daily (routine, repetitive) operations**

**事务处理支持日常(常规、重复)操作**

* **Mundane but crucial平凡但至关重要**
* **Become even more important with the growth of the internet**

1. **Definition:**

* **Collection of read/write operations**
* **Processed as one unit**
* **Reliably and efficiently processed**
* **No data loss due to interference and failures (operating system, program, disk, …)**

1. **OLTP Data Characteristics**

* **Characteristics of data:**
* **Transaction oriented-DML**
* **Inserts Updates Deletes**
* **May be inconsistent and incomplete**
* **Data may not be in its final form数据可能还没有最终形式**
* **Volatile – continually changing**
* **Data maybe subject to change**
* **Current**
* **Data related to the operation of the business TODAY!**

1. **Databases are great, BUT …**
2. **Too many of them**

* **Everybody wanted one, or two, or more**
* **Production, Marketing, Sales, Accounting**

1. **Everybody got what was best for them**

* **IBM, Oracle, Access, Microsoft**

1. **Eventually this re-created the problem databases were meant to solve**

**最终，这重新创建了数据库要解决的问题**

* **Duplicated data**
* **Inaccessible data无法访问数据**
* **Inconsistent data**

1. **What can be done about it? SPOT!**
2. **Need an integrated way of getting the ENTIRE organisational data**

**需要一个集成的方式来获得整个组织的数据**

1. **Need an Informational Database, rather than a Transactional Database**

* **A single database that allows all of the organisations data to be stored in a form that can be used to support organisational decision processes**

**一个单一的数据库，允许所有组织数据以一种可用于支持组织决策过程的形式存储**

1. **A centralised repository for decision making一个集中的决策库**

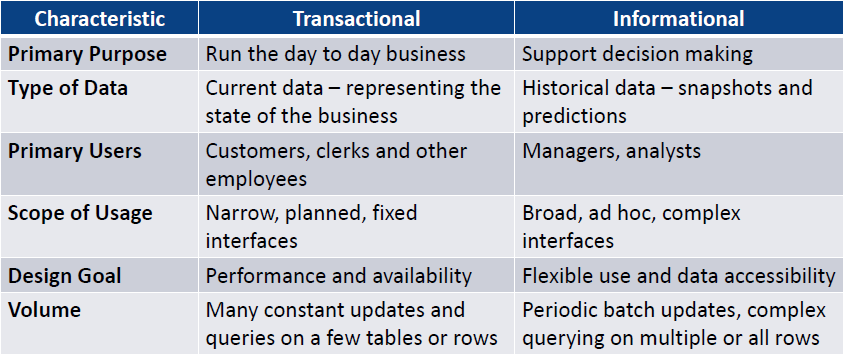
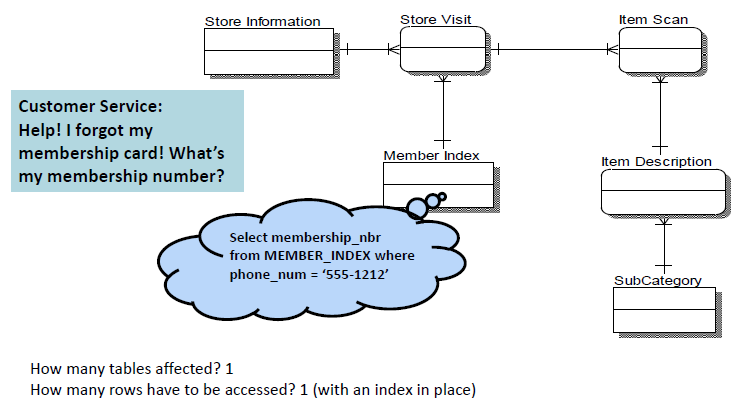
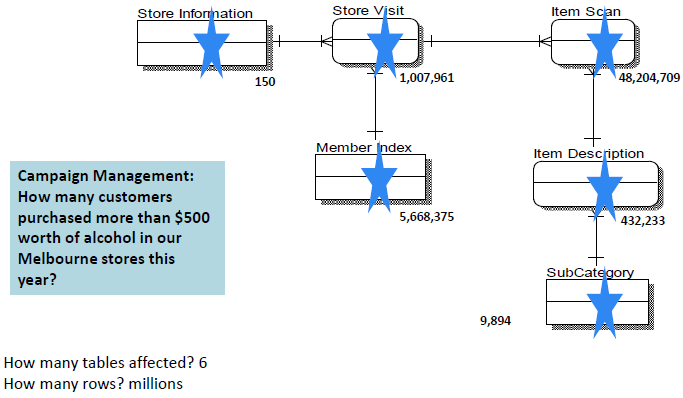
* **Populated from operational databases and external data sources**
* **Integrated and transformed data整合和转换数据**
* **Optimised for reporting优化报告**

1. **Single Point of Truth (SPOT) about the data关于数据的单点真相(SPOT)**
2. **Data Warehouse: An Informational Database**
3. **Data Warehouse:**

* **A single repository of organisational data组织数据的单一存储库**
* **Integrates data from multiple sources集成来自多个数据源的数据**
* **Extracts data from source systems, transforms, loads into the warehouse从源系统提取数据、转换数据、装入数据仓库**
* **Makes data available to managers/users**
* **Supports analysis and decision-making**

1. **Involve a large data store (often several Terabytes, Petabytes of data)**

**涉及大型数据存储(通常是数tb、pb级的数据)**

1. **Difference between Transactional and Informational Systems**
2. ****
3. **Transactional (Operational) Questions**
4. ****
5. **Analytical Questions**
6. ****
7. **DW Supports Analytical Queries**
8. **A manager may be interested in numerical aggregations**

**经理可能对数字聚合感兴趣**

* **How many?**
* **What is the average?**
* **What is the total cost?**

1. **A manager may be interested in understanding dimensions大小**

* **Sales by state by customer type**
* **Sales by product by store by quarter**

1. **Characteristics of a DW**
2. **Subject oriented**

* **Data warehouses are organised around particular subjects (sales, customers, products)**

1. **Validated, Integrated data验证、综合数据**

* **Data from different systems converted to a common format: allows comparison and consolidation of data from different sources**

**将来自不同系统的数据转换为通用格式:允许对来自不同来源的数据进行比较和合并**

* **Data from various sources validated before being sent to a data warehouse来自不同来源的数据在发送到数据仓库之前进行验证**

1. **Time variant**

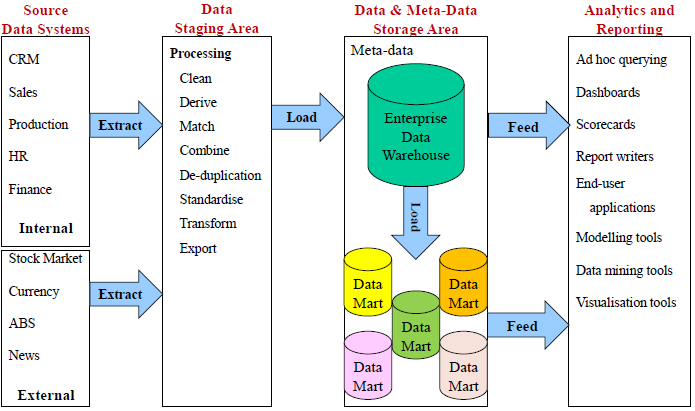
* **Historical data**
* **Trend analysis crucial for decision support: requires historical data**
* **Data consists of a series of “snapshots” which are time stamped**

1. **Non-volatile非挥发性**

* **Users have Read access only – all updating done automatically by ETL\* process and periodically by a DBA**

1. **Problems**

* **Incomplete Errors**
* **Missing Fields**
* **Records or Fields that, by design, are not recorded, e.g. the type of people that buy Big Issue from Big Issue Vendors when a sale is made根据设计，没有记录的记录或字段，例如，在进行销售时，从大问题供应商那里购买大问题的人的类型**
* **Incorrect Errors**
* **Wrong data entered into source system**
* **E.g. manual entering of data will always have a percentage of incorrect data → human error**

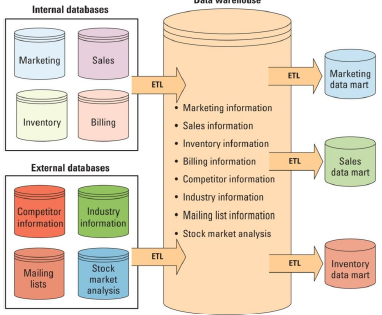
1. **A DW Architecture**
2. ****
3. **Data marts and data mining数据集市和数据挖掘**
4. **Data mart**

* **contains a subset of data warehouse information**

1. **Data-mining**

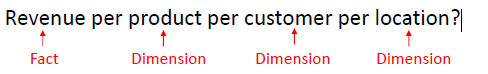
* **A process in which algorithms are applied to information to uncover patterns and relationships otherwise difficult to find**

**将算法应用于信息，以发现模式和关系，否则很难发现的过程**

1. ****
2. **Dimensional Modelling三维建模**
3. **Business Analyst World**

* **How much revenue did the product G generate in the last three months, broken down by month for the south eastern sales region, by individual stores, broken down by**
* **promotions, compared to estimates and to the previous version of the product**
* **Analysis starts usually with a single indication of something strange, then goes deep into the data, left to a new dimension, right to another, up to the summary, back down and left and right again, until the problem is identified…**

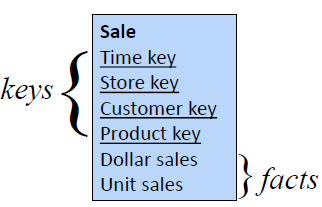
**分析通常从一个奇怪的迹象开始，然后深入数据，从左到新维度，从右到另一个维度，直到总结，再从左到右，直到发现问题……**

* **Dimensional Analysis: To support business analysts view**
* ****

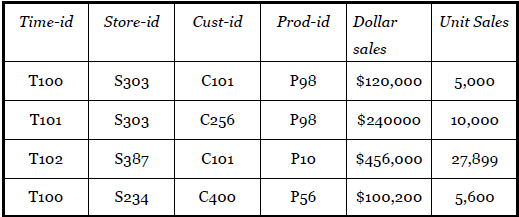
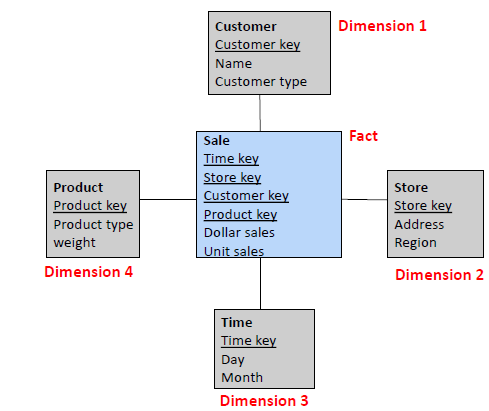
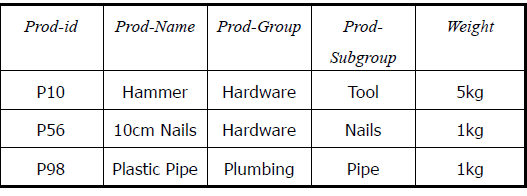
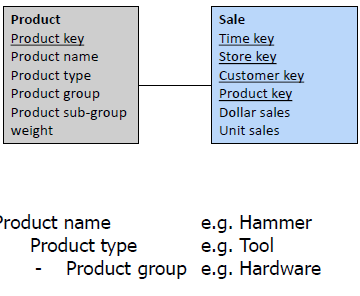
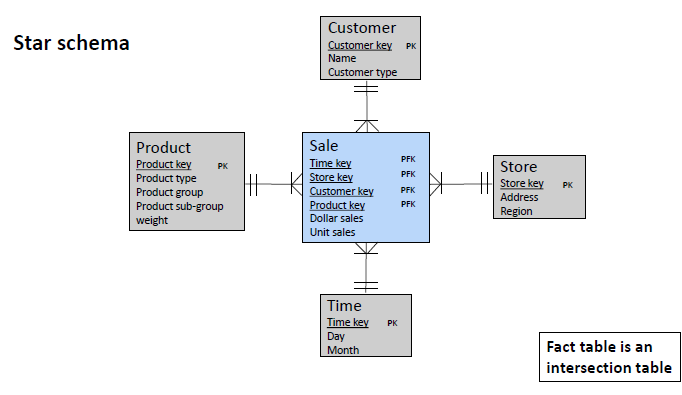
1. **Introduction to Dimensional Modelling**

* **Based on the multi-dimensional model of data and designed for retrieval-only databases**

**基于多维数据模型，专为检索型数据库设计**

* **Very simple, intuitive, and easily-understood structure**
* **Also known as star schema design也称为星型模式设计**
* **A dimensional model consists of:**
* **Fact table**
* **A fact table contains the actual business measures (additive, aggregates), called facts**
* **The fact table also contains foreign keys pointing to dimensions**
* ****
* **Actual data might look like this**
* **Finest level of detail for a fact table, determined by the finest level of each dimension**

**事实表的最优细节级别，由每个维度的最优级别决定**

* ****
* **Several dimensional tables**
* **Star schema – dimensional model**
* ****
* **Dimension Table – example**
* ****
* **(Sometimes) hierarchies in the dimensions (有时)维度中的层次结构**
* **Dimension Hierarchies**
* ****
* **Dimensional model as an ER model**
* ****

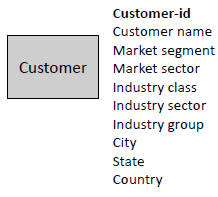
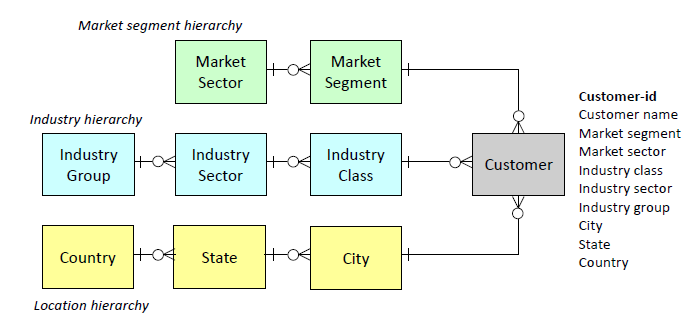
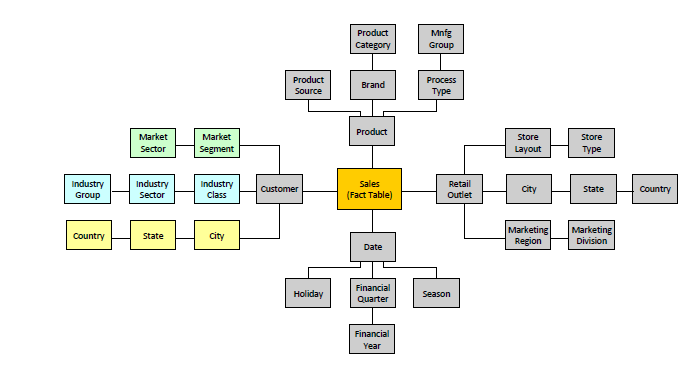
1. **Designing a Dimensional Model**

* **Choose a Business Process**
* **Choose the measured facts (usually numeric, additive quantities)**

**选择被测量的事实(通常是数字的，附加的数量)**

* **Choose the granularity of the fact table选择事实表的粒度**
* **Choose the dimensions**
* **Complete the dimension tables**

1. **Embedded Hierarchies in Dimensional Tables维度表中的嵌入式层次结构**

* ****
* ****
* ****

1. **Design Outcomes: Normalised or Denormalised?**

* **Normalisation**
* **Eliminates redundancy消除冗余**
* **Storage efficiency存储效率**
* **Referential Integrity参照完整性**
* **Denormalisation**
* **Fewer tables (fewer joins)**
* **Fast querying快速查询**
* **Design is tuned for end-user analysis设计是为最终用户分析而调优的**

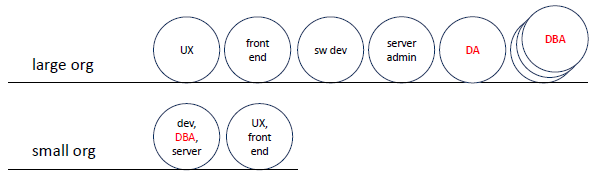
1. **Exercise**

* **We are making a data warehouse for a real estate agency. The company wants to track information about the selling of their properties. This warehouse keeps information about the agents (license#, first name, last name, phone #), buyers that come in (buyer id, first name, last name, phone #), and property (property#, property address, price). The information managers want to be able to find is the number of times a property is viewed, sales price. The information needs to be broken down by rental agent, by buyer, by property and for different time (day, week, month, quarter and year).**

**Database Architecture and Administration**

1. **The DBA role**
2. **Primarily concerned with “maintenance” / “ops” phase but should be consulted during all phases of development**

**主要涉及“维护”/“运维”阶段，但在开发的所有阶段都应征求意见**

1. **“Database Administrator” or “DBA” often framed as a “job” or a “person”**
2. **Large companies-many DBA’s**
3. **Small company developer is the DBA**
4. **DBA role can be made redundant by Cloud based DBMS or “data as a service” DAAS (often IAAS or PAAS)**
5. ****
6. **Data and Database Administration**
7. **Data Administrator (CDO / CSO) (management role)**

* **data policies, procedures and standards**
* **planning**
* **data conflict resolution**
* **managing info repository管理信息存储库**
* **internal marketing & education**
* **Compliance with legislation (EU GDPR AUS Privacy Act)**
* **Compliance with company policy (e.g. Unimelb privacy policy)**

1. **Database Administrator (technical role)**

* **analyze and design DB**
* **select DBMS / tools / vendor**
* **install and upgrade DBMS**
* **tune DBMS performance调优数据库管理系统性能**
* **manage security, privacy, integrity**
* **backup and recovery**

1. **DA Versus DBA**
2. ****
3. **Architecture of a Database Management System (DBMS)**
4. **Database Systems Architecture**

* **A Database Management System (DBMS) exists as one entity in two places**
* **In Memory**
* **Physically on disk**
* **Both places manage**
* **Data (the reason we have the DBMS)**
* **Performance (how it performs as it is used & grows)**
* **Concurrency (manages high volumes of users)** **并发性(管理大量用户)**
* **Recoverability (assist in recovery and availability)**

**可恢复性(协助恢复和可用性)**

* **One place is persistent the other transient**

**一个地方是持久的，另一个地方是短暂的**

* **Disk representation is always present磁盘表示总是存在的**
* **Memory transient only exists when DBMS is running**

**内存瞬态只在DBMS运行时存在**

1. **DBMS Overview**

* ****

1. **Query Processing查询处理**

* **Parsing解析**
* **Syntax is correct and can “compile”** **语法正确，可以“编译”**
* **DBMS User Permissions**
* **Resources (Data, Code, be able to Record/Change /Retrieve**
* **Optimising**
* **Execution Plan and Execution Cost**
* **Evaluate indexes, table scans, hashing计算索引，表扫描，哈希**
* **Eliminate worst, consider best options**
* **Lowest cost theoretically “best”**
* **Execution**
* **Meet the ACID test (atomicity, consistency, isolation, and durability),**
* **Atomic: All rows succeed, or all fail**
* **Ensure resources are available**
* **Data, Log changes, Memory, Cursor to do the work for the USER**

1. **Concurrency Control并发控制**

* **Manages the work of the DBMS.**
* **Transaction Manager handles all aspects of the SQL transaction which DBMS user wants WHAT resource**
* **Lock Manager is a list of what resources are locked and by which user at what level ( and who is waiting)**
* **Not only tables, indexes**
* **buffers, cursor, memory addresses of resources**

**缓冲区、游标、资源的内存地址**

* **Essential to manage large scalable DBMS**

**对于管理大型可伸缩DBMS至关重要**

* **Enables 1,000,000s of concurrent users**
* **Like a Traffic Policemen controlling the flow of traffic**
* **Who can do what (allowed to do what they need to do)**
* **Who has to wait (queue)** **谁要等候(排队)**
* **Who can travel through the intersection concurrently**

**谁可以同时穿过十字路口**

* **Usually readers of data**
* **What transactions have completed, in progress, compiled**编辑**.**
* **What resources are involved with that transaction**
* **Who last used, is using and wants those resources**
* **SQL, Cursor, Index, Table, Rows, File Access, Recovery Logs**

1. **Storage**

* **File and Access Methods**
* **Disk to Memory to Disk**
* **Read a buffer or a block of buffers**
* **Buffer Pool**
* **Data in memory**
* **Row data**
* **Index data**
* **Organised**
* **Disk Space Management**
* **How to organise growth of data on disk efficiently by writing efficiently.**

1. **Storage-File Management**

* **How to access the file**
* **Full Scan (full table scan)**
* **From beginning to end of the entire file**
* **Partial Scan (index range)** **部分扫描(索引范围)**
* **Using an index to scan a range of values, e.g. (2 < deptno <9)**
* **Page only (file header and page)** **仅限页面(文件头和页面)**
* **Read the index and data file**

1. **Storage-Buffer Pool and Disk Space**

* **Buffer Pool**
* **Table data and indexes are read from disk into the buffer (dedicated area of memory)**
* **May contain multiple copies of the same data**
* **Need to know which copy is the current committed version**
* **Disk Space Management**
* **How to allow files to grow on disk (and set max growth size)**
* **File organization**
* **e.g. index reorganisation;**
* **varchar growth, e.g. Brown (5) grows to Nicholson (9) for last name**

1. **Storage-Disk Space Management**

* **Hierarchical Structure层级结构**
* **Storage of Object (tables, indexes, rollback, logs)**
* **Row by Row space management-inefficient**
* **Page better (virtual block, 4K and 2K are the most commonly used sizes).**
* **Block Best (but each device may support a different block size)**

1. **Buffer Pool**

* **Many Object Types (Tables, Indexes, Undo)**
* **Each buffer contains rows, b+ tree leaf etc.**
* **Each buffer can have one of four status types:**
* **Current**
* **In use current committed version of data (row)**

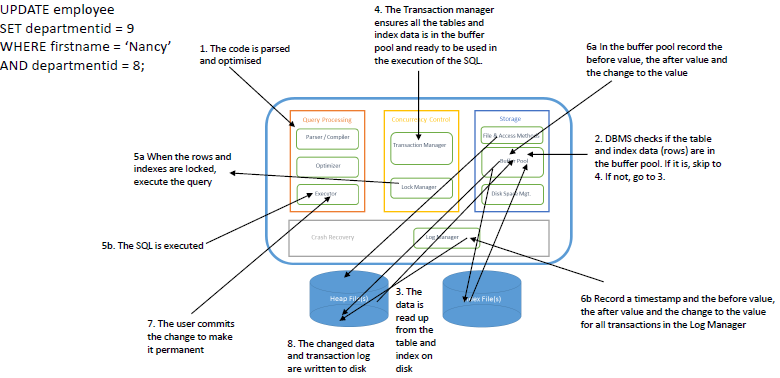
**正在使用数据(行)的当前提交版本。**

* **Active**
* **Most recent change (may not be committed)**
* **COMMIT (Current: DepartmentID=9)**
* **Stale不新鲜的**
* **An old version of the data**
* **Aged**
* **Old and about to be removed from buffer pool**

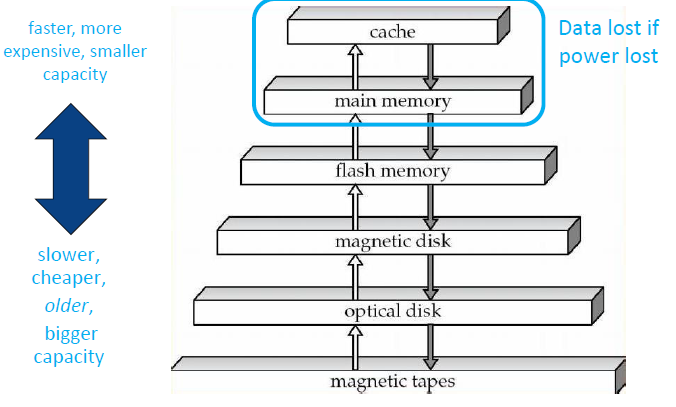
1. **Log Manager**

* **Recovery**
* **Log Manager records ALL changes**
* **Statement**
* **Transaction**
* **Statement**
* **Rollback values**
* **Before and After values**
* **Timestamp begin**
* **transaction, savepoint \* and commit timestamps**
* **Database**
* **Data Dictionary Changes**

1. **DBMS-How a transaction works**

* ****

1. **Storage media hierarchy**

* ****

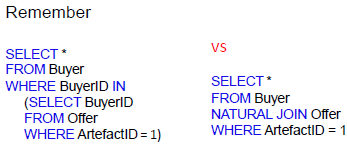
1. **What affects database performance?**

* **Caching data in memory, e.g. data buffers**

**在内存中缓存数据，例如数据缓冲区**

* **Data and Code found in memory**
* **Avoids a read from Disk**
* **Reads are expensive**
* **Goal in to minimize reads (and writes)**
* **Writes are necessary (recovery logs, changed data)**
* **“in memory databases”**
* **All code and all data loaded into memory on database start and stays until shutdown**

**所有代码和数据在数据库启动时加载到内存中，并一直保持到关闭**

* **Placement of data files across disc drives跨磁盘驱动器放置数据文件**
* **Fast storage such as SSD**
* **Spread the files across the physical server跨物理服务器传播文件**
* **RAID (0, 1, 5/6)**
* **We can’t avoid writes**
* **Spread files across many disks**
* **Avoid contention避免争用**
* **(many users competing for same resource)**
* **Recovery Logs (always writing)**
* **faster disk**
* **SSD (Solid State Drives)**
* **No moving parts nothing to break down**
* **Faster I/O (compared to other disk types)**
* **Database replication and server clustering数据库复制和服务器集群**
* **Distributed data**
* **Spreads the load传播的负载**
* **Data kept only where it is needed**
* **Less work per physical server-faster response times**
* **Replicated Data**
* **Spreads Load**
* **Less work per physical server-faster response times**
* **Use of indexes to speed up searches and joins**
* **Good choice of data types (especially PKs)**
* **Good program logic (no long running CRUD增删改查)**
* **Good query execution plans**
* **The best execution plan has the lowest “cost”**
* **Known as Cost Based Optimization (CBO)** **所谓的成本优化(CBO)**
* ****
* ****
* **Good code (no deadlocks)**
* ****

1. **Indexing**
2. **Indexes**

* **If the index did not exist, DBMS would have to view each record in the employee table sequentially to find a matching employee id.**

**如果索引不存在，DBMS必须依次查看员工表中的每条记录，以找到匹配的员工id。**

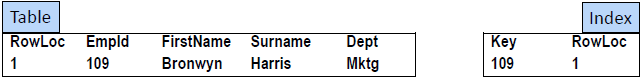
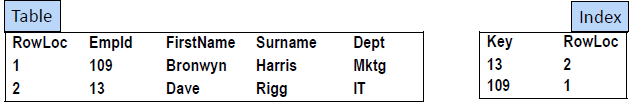
* **With 100,000 records, that could be a long time (A second or two).**
* **If there were 10 million rows, the delay would increase.**
* **The more rows we have…. the slower the sequential search.**
* **Suppose we create an index on the Employee Id of the Employee table**
* **An index stores**
* **The index value (the employee id)**
* **The location of the record within the database**
* **Whenever you create a Primary key, DBMS automatically creates a Primary Key index.**

1. **Sequential Record Access顺序记录访问**

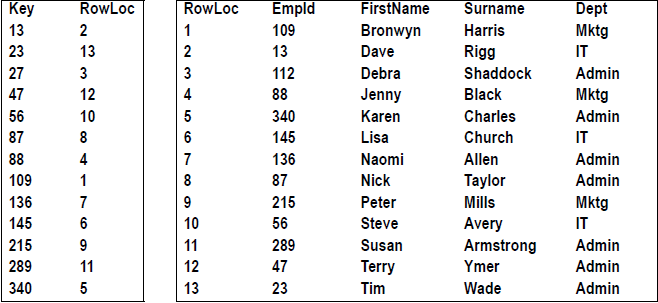
* **Read each record one by one from the beginning of the table until the required record is found.**
* **The DBMS may have to evaluate 50,000 separate records.**
* **Assume that each record is very large.**
* **Assume each evaluation requires one disk I/O. \***
* **Disk I/O is the slowest computer activity.**
* **On average, a search for an employee will require 25,000 disk Os**
* **In reality, a DBMS will read records as blocks of data (e.g. 2K). Therefore many records can be read into memory with a single disk I/O**

**实际上，DBMS会以数据块的形式读取记录(例如2K)。因此，许多记录可以通过单个磁盘I/O读入内存**

1. **Inserting data into a simple Index**

* **When a table has a Primary Key, an Index based on that Primary Key is automatically created.**
* **As each row is added, the DBMS automatically inserts the search key and the row location information into the index file**
* **Insert 1st row. Employee 109**
* **A row is inserted into the Employee table**
* **The search key and row location is added to the PK index**
* ****
* **Insert 2nd row. Employee 13**
* **A row is inserted into the Employee table The row is added to the table.**
* **The search key and row location is added to the PK index**
* **The PK automatically maintained in PK sequence**
* ****

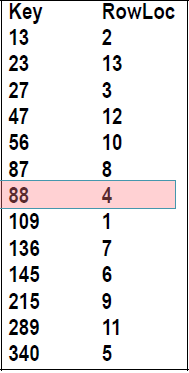
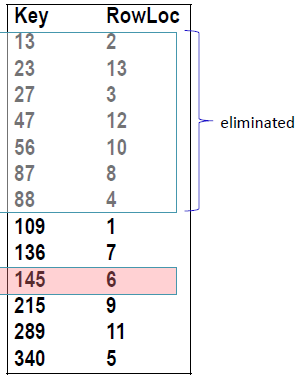
1. **The Index**

* **Obviously, the Index is much smaller than the table data**
* **Searching the index is quicker than sequentially searching the table**
* **The Index is in key sequence. This makes searching quick.**
* ****

1. **Searching the Index**

* **Locating the row for employee 289 is quicker using the index. A RDBMS may use a search method on the index such as a Binary Search**

**使用索引可以更快地定位员工289的行。RDBMS可以在索引上使用一种搜索方法，例如二进制搜索**

* **A Binary Search continues to split the index in half until a match is/is not found**
* **Start at the middleth position**
* ****
* **The value being searched for is either The current key value OR A key value higher in the list OR A key value lower in the list**
* **At least half of the key values are eliminated.**
* **The eliminated values do not need to be searched**
* **Of the remaining key values, examine the new middle th position**
* ****
* **The value being searched for is either The current key value OR A key value higher in the list OR A key value lower in the list OR**
* **Again, half of the keys are eliminated.**

1. **Advantages of using an Index**

* **Using an index is a quick method to locate rows.**
* **Large portions of the index (or the entire index) may be read into the computer's memory at once before searching for rows in a table.**

**在搜索表中的行之前，可以将索引的大部分(或整个索引)立即读入计算机的内存。**

* **Fewer disk I/Os (input/outputs the slowest function of a computer).**

**磁盘I/ o减少(输入/输出是计算机最慢的功能)。**

* **This reduces the number of disk I/ Os required to read table rows**
* **This speeds up retrieval time这加快了检索时间**
* **In the our example, finding employee 289 took 2 disk I/ Os**
* **One to read the index into RAM**
* **One to read the 11 th row in the table**

1. **Additional information**

* **The index does not contain a record number**
* **The record number we see in the datasheet doesn't really exist!**
* **Fred Blogs data may be stored in Record 3. If Record 1 & 2 are deleted, then Fred becomes Record 1.**

**Fred Blogs数据可能存储在Record 3中。如果记录1和2被删除，那么Fred变成记录1。**

* **So the Record Number is a poor way of identifying a record**
* **(That's one reason we use Primary)**
* **Instead, the RDBMS uses a physical address on disk to specify the location of the record. This is the value of RowLoc in the previous slides**

**相反，RDBMS使用磁盘上的物理地址来指定记录的位置。这是前面幻灯片中RowLoc的值**

* **In Oracle RDBMS the address is something like AAAYUzAAEAABIjLAAJ**
* **MS Access makes it almost impossible for you to view the physical address.**

1. **Indexes speed up report generation**

* **There are many IT database stories involving reports that**
* **Use many tables with Foreign Key lookups使用许多带有外键查找的表**
* **Involve millions or billions of database records**

**涉及数百万或数十亿条数据库记录**

* **Take hours to produce**
* **A daily report that takes more than 10 hours to run may be next to useless.**
* **Such reports do exist!**
* **Creating indexes on fields central to the report have often reduced report generation time from many hours to a few minutes.**

**在报表的中心字段上创建索引通常可以将报表生成时间从几个小时缩短到几分钟。**

1. **The cost of creating Indexes**

* **Why not index every field?:**
* **Disk Space a minor consideration. Disk space is cheap.**

**磁盘空间是次要考虑事项。磁盘空间很便宜。**

* **Update Time a major consideration.**
* **When records in a table are inserted, changed, deleted:**
* **Every index must be updated**
* **so that the index remains in key sequence**
* **Each change takes time to complete**
* **Too many indexes can make unacceptable delays**

**过多的索引会造成不可接受的延迟**

* **Imagine each Unit that you enrol in, causes a few seconds delay**
* **as all of the indexes are updated**
* **Imagine each time Australia Census data is recorded there is a delay as all indexes are updated**
* **Users will not be happy**

1. **Deletion / Recreation of Indexes?** **删除/重新创建索引?**

* **Database management in an organisation often involves compromises**

**和解**

* **An organisation may run a once-a-week report.**
* **The report requires many indexes to run acceptably**
* **It may make sense to**
* **Create the indexes prior to running the report**
* **it make take a few minutes for each index to be created**
* **Run the report**
* **Drop the indexes**
* **Often such tasks can be scheduled to automatically take place**

1. **DBMS Optimisers**

* **DBMS systems have optimisers最佳化**
* **Optimisers are designed to decide the quickest and most efficient way of accessing data.**
* **Multiple indexes may exist, but it is the DBMS optimiser that decides if it is of benefit to use the indexes.**
* **Image we have a table with a small number of rows.**
* **We have indexes based on the Employee Id and Surname.**
* **Imagine that it only takes one or two reads to load the entire table into memory,**
* **It may be quicker to read/load all rows into memory**
* **compared to reading / loading the Indexes into memory and then retrieving the appropriate row(s)**

**与读取/加载索引到内存，然后检索适当的行相比**

* **Optimisers attempt to reduce the time taken to process the query.**

1. **When to create indexes**

* **For each table, choose the columns you will index:**
* ***queried frequently* (used in WHERE clauses)**
* **used for *joins* (PK to FK)**
* **primary *keys* (automatic in most DBMS)**
* **foreign *keys* (automatic in MySQL)**
* **unique columns (automatic in most DBMS)**
* **Large tables only-small tables do not require indexes**
* **if you frequently retrieve less than about 15% of the rows**
* **Wide range of values (good for regular indexes).**
* **Small range of values (good for bitmap/hash indexes).**

**小范围的值(适用于位图/哈希索引)。**