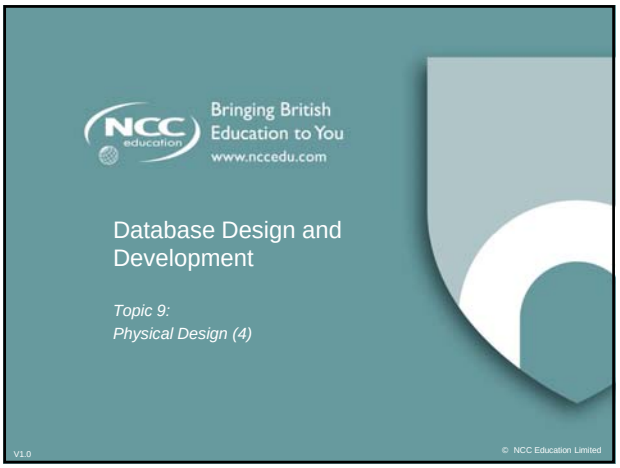
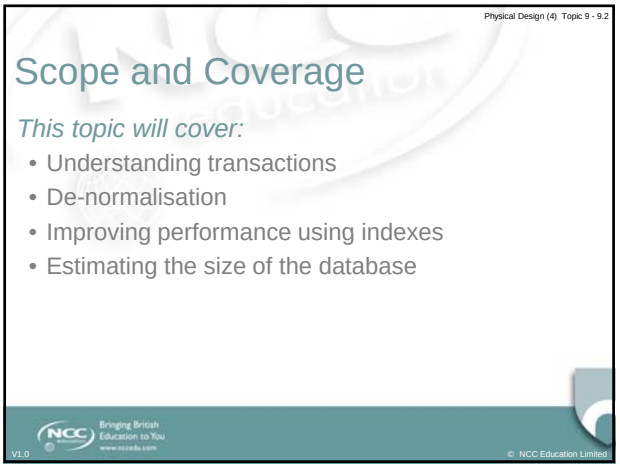
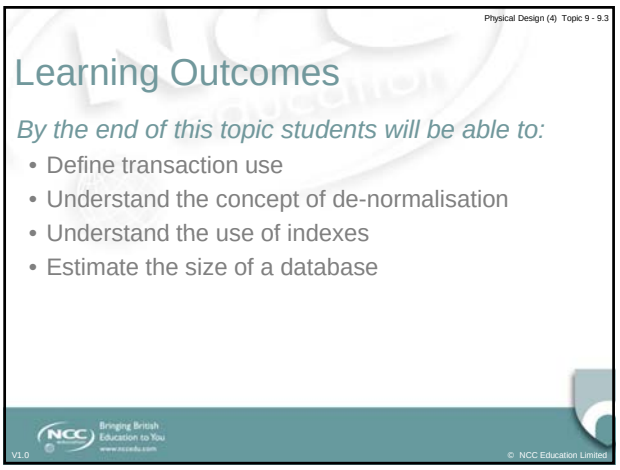
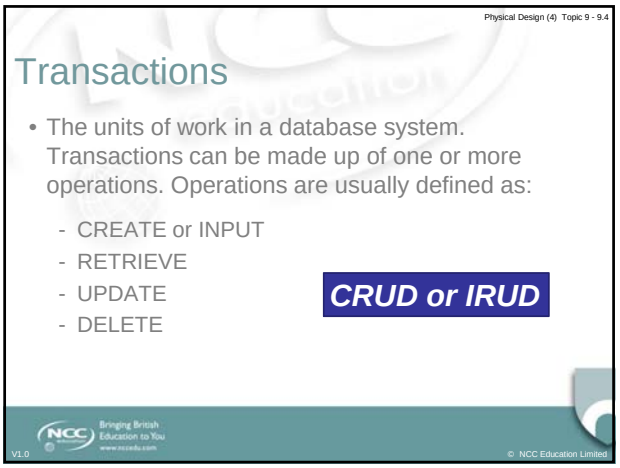
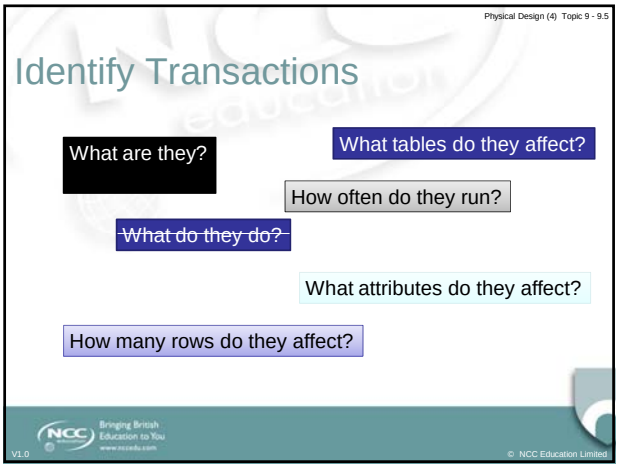
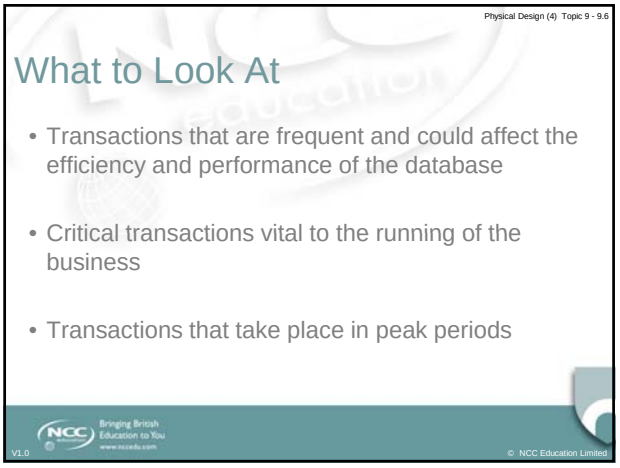

Bringing British
Education to You
www.nccedu.com
Database Design and Development
Topic 9:
Physical Design (4)
V1.0
© NCC Education Limited


Physical Design (4) Topic 9 - 9.2
Scope and Coverage
This topic will cover:
• Understanding transactions
• De-normalisation
• Improving performance using indexes
• Estimating the size of the database
V1.0
© NCC Education Limited


Physical Design (4) Topic 9 - 9.3
Learning Outcomes
By the end of this topic students will be able to:
• Define transaction use
• Understand the concept of de-normalisation
• Understand the use of indexes
• Estimate the size of a database
V1.0
© NCC Education Limited


Physical Design (4) Topic 9 - 9.4
Transactions
• The units of work in a database system. Transactions can be made up of one or more operations. Operations are usually defined as:
- CREATE or INPUT
- RETRIEVE
- UPDATE
- DELETE
CRUD or IRUD
V1.0
© NCC Education Limited


Physical Design (4) Topic 9 - 9.5
Identify Transactions
What are they?
What tables do they affect?
How often do they run?
What do they do?
What attributes do they affect?
How many rows do they affect?
V1.0
© NCC Education Limited

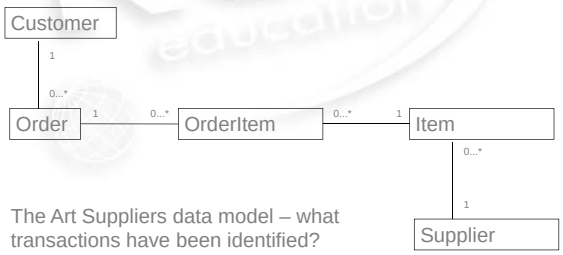

Physical Design (4) Topic 9 - 9.6
What to Look At
• Transactions that are frequent and could affect the efficiency and performance of the database
• Critical transactions vital to the running of the business
• Transactions that take place in peak periods
V1.0
© NCC Education Limited

How to Investigate

- Look at each transaction and work out what tables it will affect.
- Work out if there are tables that are used by a lot of transactions
- Look at how the data is affected by the transaction

Analysing Transactions

- Does the transaction involve any predicates (specific conditions in the where clause).
- CRUD
- Attributes?
- Frequency?



The Art Suppliers data model – what transactions have been identified?

Transaction ID	Details
A	Create a new customer
B	Create a new order for up to 10 items
C	Create new items
D	Create new suppliers
E	Update an order by adding or deleting an order item
F	Delete an order item
G	Update a customer's details
H	Delete a customer

Blank CRUD Matrix

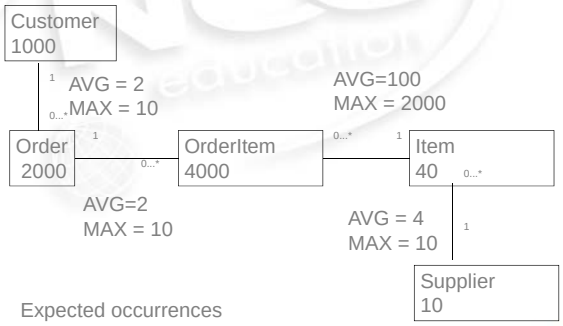
Transaction /Relations	A	B	C	D	E	F	G	H
Customer								
Order								
OrderItem								
Item								
Supplier								

Draw this diagram and then look at the transactions again.

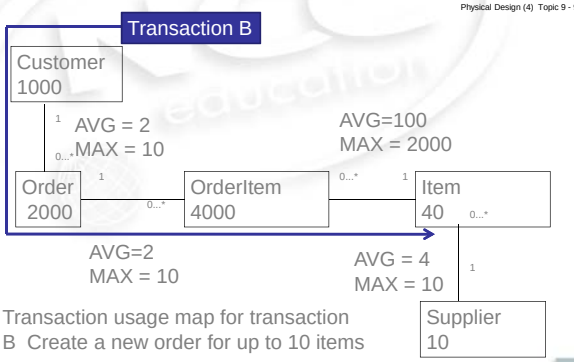
Transaction /Relations	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
Customer	**C**	**R**					**R U**	**D**
Order		**C**			**R U**			
OrderItem		**C**			**C R U D**	**R D**		
Item		**R**	**C**					
Supplier				**C**				

Attribute Level

- A transaction may affect some attributes in a table but not others.
- Or it may affect different attributes in the same table in different ways, updating one and deleting another for example.



Expected occurrences



Transaction usage map for transaction B Create a new order for up to 10 items

For Each Transaction

- The tables and columns accessed and type of access (CRUD)
- Columns used in any search condition
- Any join conditions for queries
- Expected frequency of transaction
- Performance goals

Transaction Analysis Form
1st Jan 2010
Transaction (b) Create a new order for up to 10 items
Transaction volume
Average 50 per day
Peak 100 times per day during start of new term at the Art College

SELECT CUSTOMERID FROM CUSTOMERS WHERE CUSTOMER NAME = <Customer making order>
INSERT INTO ORDERS VALUES (:NEW_ORDER_ID, :CUSTOMERID, :TODAYS_DATE)
SELECT ITEMID FROM ITEMS WHERE ITEMNAME = <Item customer is ordering>
INSERT INTO ORDERITEMS VALUES (:NEW_ORDER_Id,:ITEMID)

Access	Entity	Type of Access	Average Number PER HOUR: 10	Peak Number PER HOUR 50
1	CUSTOMER	R		
2	ORDER	C		
3	ITEM	R		
4	ORDERITEM	C		

Performance

- Increasingly databases contain large amount of data...
- The rate at which a query can return an answer can be slowed when it has to sort though large numbers of records.
- Performance becomes an issue...

Indexes

- Improve performance.
- They work by creating entries in a special structure that makes it easier to find a record

Choose Indexes

- The columns used most often in joins
- Columns that are used for ordering

Index Types

- Primary index
- Secondary index
- Clustering index

Primary Index

- Built around a key field that is used for ordering. A unique value for every entry in the index.

Secondary Index

- Defined on a non-ordering field
- May not contain unique values
- Improves the performance of queries that use columns other than primary key

Use of Secondary Indexes

- Mechanism for specifying additional key columns.
- For example Customer would be searched often on Customer Name as well as the primary key and so a secondary index could be used on it.

Clustering Index

- Built around non-key column or columns.
- So there can be more than one record corresponding to the indexed column
- The non-key field is called the 'clustering field'

Examples of Creating Indexes in SQL

- To create primary index
  - CREATE UNIQUE INDEX CustomerIDIndex
  - ON Customer(CustomerID)
- To create clustering index
  - CREATE INDEX OrderDateIndex
  - ON Order(OrderDate) CLUSTER
- To create secondary index use CREATE INDEX syntax without specifying unique.
- NOTE: Index creation is not yet standard SQL

Overheads of Use of Indexes

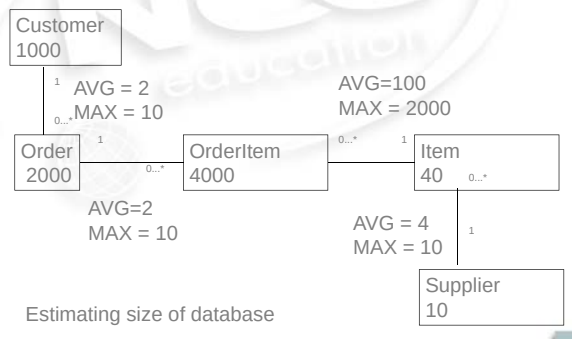
- A record is added to the index table every time a new record is added to the table where there is a secondary index.
- Updating the indexed record means an update to the index table
- More disk space need to store index tables
- Impact on performance if indexes are all consulted every time a query is run

De-normalisation

- We have created a database following all the rules of normalisation...
- Now we can break them to make the database work quicker and perform better...

Candidates in Art Supply for De-Normalisation

- If most enquires are about orders but also require some customer information then we might de-normalise by including the customer_name on the Orders table.
- We could also use a trigger to allow creation of Orders with customer details on the Orders table.



Estimating size of database

Size and Growth

- Initial number of rows
- Daily, Weekly, Monthly and Yearly growth rates
- Peak periods
- Deletion and archiving policy

The Development Process in Conclusion

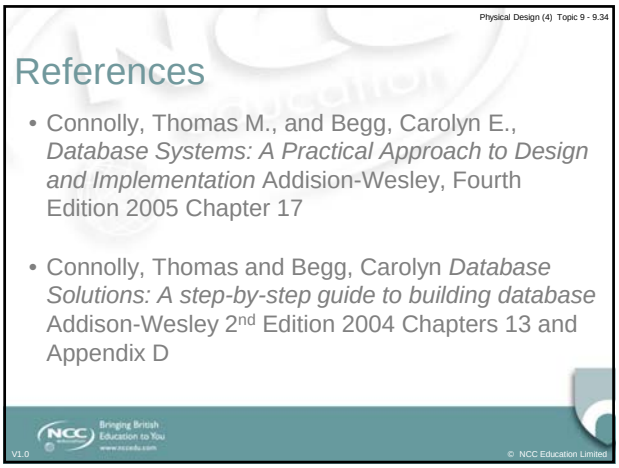
- Data loading
- Maintenance
- Expansion
- Change
- Lifespan
- Is development ever complete?

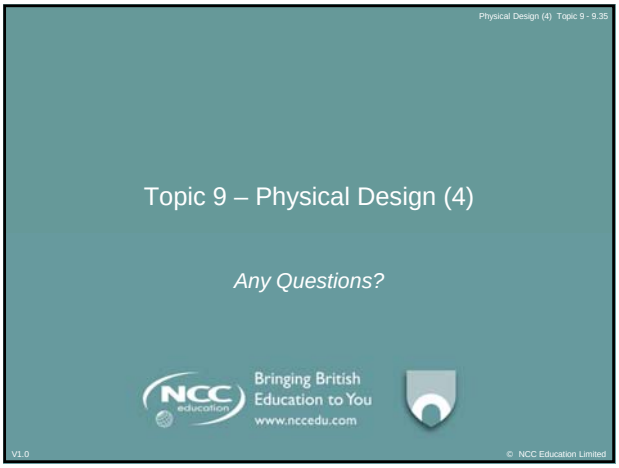
Learning Outcomes

By the end of this unit students will be able to:

- Define transaction use
- Understand the concept of de-normalisation
- Understand the use of indexes
- Estimate the size of a database

Have we met them?


Physical Design (4) Topic 9 - 9.34
References
• Connolly, Thomas M., and Begg, Carolyn E., *Database Systems: A Practical Approach to Design and Implementation* Addision-Wesley, Fourth Edition 2005 Chapter 17
• Connolly, Thomas and Begg, Carolyn *Database Solutions: A step-by-step guide to building database* Addison-Wesley 2nd Edition 2004 Chapters 13 and Appendix D
Bringing British Education to You
V1.0
© NCC Education Limited


Physical Design (4) Topic 9 - 9.35
Topic 9 – Physical Design (4)
Any Questions?
Bringing British Education to You
www.nccedu.com
V1.0
© NCC Education Limited