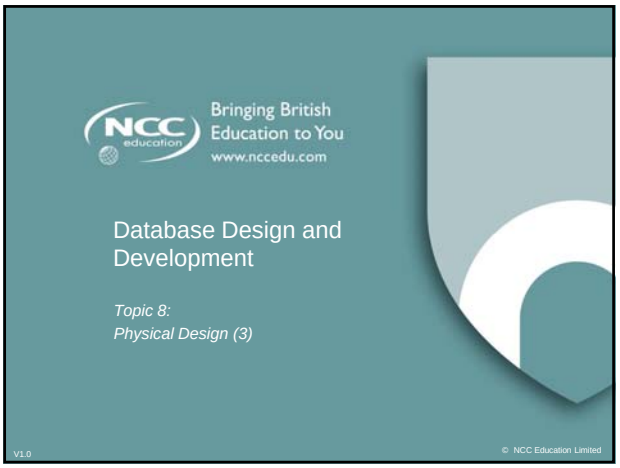
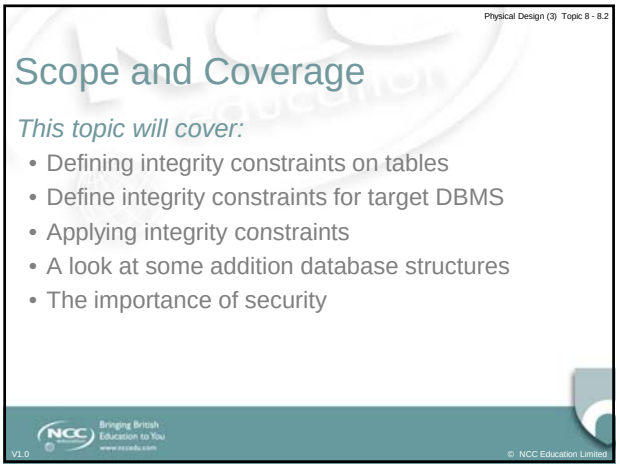
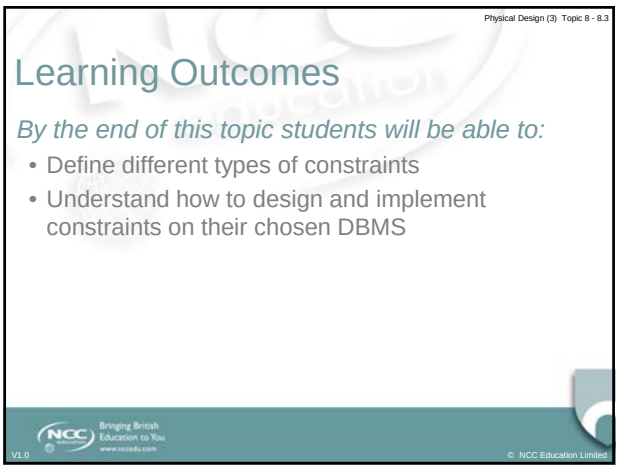

Bringing British
Education to You
www.nccedu.com
Database Design and Development
Topic 8:
Physical Design (3)
V1.0
© NCC Education Limited


Physical Design (3) Topic 8 - 8.2
Scope and Coverage
This topic will cover:
• Defining integrity constraints on tables
• Define integrity constraints for target DBMS
• Applying integrity constraints
• A look at some addition database structures
• The importance of security
V1.0
© NCC Education Limited


Physical Design (3) Topic 8 - 8.3
Learning Outcomes
By the end of this topic students will be able to:
• Define different types of constraints
• Understand how to design and implement constraints on their chosen DBMS
V1.0
© NCC Education Limited

Physical Design (3) Topic 8 - 8.4

Types of Constraints

- Entity integrity
- Referential integrity
- Propagation constraints
- Domain constraints
- Table constraints

Physical Design (3) Topic 8 - 8.5

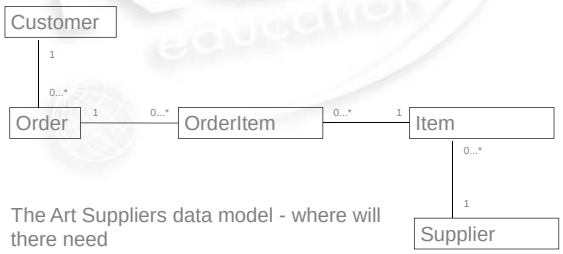
Activity – Recap - 1

- What is the Entity integrity rule?
- What is referential integrity?

Physical Design (3) Topic 8 - 8.6

Activity – Recap - 2

- What is the Entity integrity rule?
  - The primary key of an entity cannot contain nulls.
- What is referential integrity?
  - If a foreign key contains a value then that value must refer to an existing tuple in the source table



The Art Suppliers data model - where will there need
to be constraints?

Source or Parent Table

```
Create table customers
(CustomerNo integer(5) not null,
first_name varchar(30),
last_name varchar(30),
primary key emp_no);
```


Referencing or Child Table

```
Create table Orders
(OrderID   integer(5) not null,
CustomerID integer(5) not null,
OrderDate datetime,
primary key OrderID,
foreign key (CustomerID) references Customer
(CustomerID);
```

Referential Integrity Constraint – Another Example

```
Create table workers
(emp_no integer(5) not null,
first_name varchar(30),
last_name varchar(30),
job_title varchar(30),
age integer(3),
dept_no integer(5),
primary key emp_no,
foreign key (dept_no) references Departments (dept_no)
```


Propagation Constraint

- What happens if we delete a Item from our Art Supply database?
- There are lots of OrderItem records that reference it. What happens to them?

Table with Propagation Constraint

```
Create Table Item
(ItemID integer NOT NULL,
SupplierID integer NOT NULL,
Price float,
Primary Key (ItemID),
Foreign Key (SupplierID) REFERENCES Supplier(SupplierID),
On delete no action
On update cascade);
```

Options for Propagation

- No action
- Cascade
- Set Default
- Set Null

Domain Constraints

- Product Type could be enforced as...
  - a check constraint
  - separate domain using Create Domain statement
  - as a foreign key to another table

Check Constraint

```
Create Table ProductType
(ProductTypeID integer NOT NULL,
ProductTypeName varchar (30) NOT NULL,
ProductTypeColour varchar (20),
Primary Key (ProductTypeID)
Check (ProductTypeColour in 'Red','Blue','Green'));
```

As a Separate Domain

Create Domain ProductTypeColour As varchar(20)
Default 'Red'
Check (Value in ('Red','Blue','Green'));

- The table'ProductType' will set the ProductTypeColour attribute as this domain ProductTypeColour

```
Create Table ProductType
(ProductTypeID integer NOT NULL,
ProductTypeName varchar (30) NOT NULL,
ProductTypeColour BoatType,
Primary Key (ProductTypeID));
```


As a Separate Table

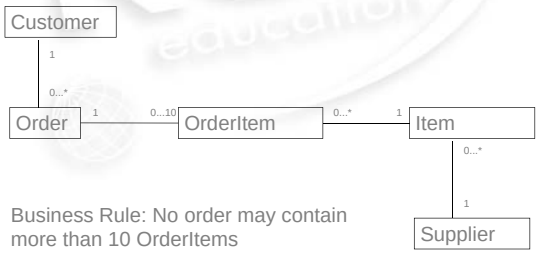
```
Create Table ProductTypeColour
(ProductTypeColourCode Varchar(3),
ProductTypeColourDescription Varchar(20)
Primary Key (ProductTypeColourCode));
```

With the corresponding Foreign Key in Boat

```
Create Table ProductType
(ProductTypeID integer NOT NULL,
ProductTypeName varchar (30) NOT NULL,
ProductTypeColour varchar (3),
Primary Key (ProductTypeID)
Foreign Key (ProductTypeColourCode)
References ProductTypeColour (ProductTypeColourCode);
```


Business Rules Enforced by Constraints

- Business rules are derived from requirements analysis and documented in logical design
- They depend on the operations of the 'real world' business



Business Rule: No order may contain more than 10 OrderItems

Table Constraints

```
Create Table Orders
(OrderID integer NOT NULL,
CustomerID integer NOT NULL,
OrderDate datetime NOT NULL
Constraint MaximumOrderItems
Check(Not Exists(Select OrderID
From OrderItems
Group By OrderID
Having Count(*) >10)),
Primary Key (OrderID)
Foreign Key (CustomerID) REFERENCES Customer(CustomerID)
On delete no action
On update cascade);
```


Table Constraints – Another Example

```
Create Table Rental
(BoatID integer NOT NULL,
CustomerID integer NOT NULL,
RentalStartDate datetime NOT NULL,
RentalEndDate datetime NOT NULL
Constraint MaximumRentals
Check(Not Exists(Select BoatID
From Rentals
Group By BoatID
Having Count(*) >10)),
Primary Key (BoatID, CustomerID, RentalStartDate)
Foreign Key (CustomerID) REFERENCES Customer(CustomerID),
Foreign Key (BoatID) REFERENCES Boat (BoatID)
On delete no action
On update cascade);
```

Triggers

- Triggers are supported by some RDBMS products
- Triggers are pieces of procedural logic / programming that can be attached to database objects
- They 'fire' (come into operation) when some database operation such as an insert, update or delete takes place

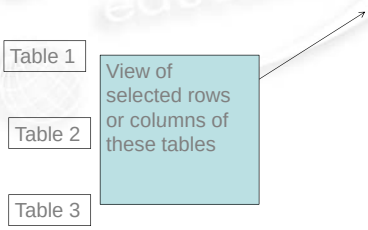
Using Triggers to Enforce Constraints

```
Create trigger order_maximum
Before insert or update on OrderItems
For Each Row
Declare
  x Number;
Begin
    Select Count(*) into x
                From OrderItems
      Where OrderID = :new.OrderID;
    IF x > 10 Then
          raise_application_error(-2000,('OrderItems' || :new.memberNo || '10
items ordered');
      End if
End;
```


Some Additional Database Structures

- Views
- Indexes
- Sequences

Improving Performance with the use of Views



Example of Creating a View

```
Create View OrderSummary as
(Select O.OrderID, O.OrderDate, I.ItemName, I.Price,
OI.Quantity, O.Total
From Orders O, OrderItems OI, Items I
Where O.OrderID = OrderItems.OrderID
And I.ItemID = OI.ItemID);
```

- What will be the purpose of this view?

Indexes

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

Sequences

- Sequential numbers that can be used to increment an ID number
- Equivalent to an auto-number in MS Access

Design Security Measures

- Why is security important?
- System security
- Data security
- 1999 ISO SQL Standard (SQL3) facilities for managing roles

Roles in a System

- Not every user is the same
- The will need to access different parts of the system and access it in different ways

SQL Facilities to Manage Roles

- Grant – gives access to an object (such as a table) to a particular role or user in the database system.
- Revoke – removes access to an object (such as a table) to a particular role or user in the database system

Grant

- Grant create on Customers to Admin
- Grant all on Customers to Manager

Revoke

- Revoke all on Customers from Admin
- Revoke delete on Customers to Manager

Learning Outcomes

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

- Define different types of constraints
- Understand how to design and implement constraints on their chosen DBMS

Have we met them?

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 12 - 13

Topic 8 – Physical Design (3)

Any Questions?

Bringing British Education to You
www.nccedu.com