SQL Server: Understanding Database Fundamentals (98-364)

INTRODUCTION

Tim Radney
PRINCIPAL CONSULTANT

@tradney www.SQLskills.com

Module Overview

Introduction to SQL Server

Versions and editions of SQL Server

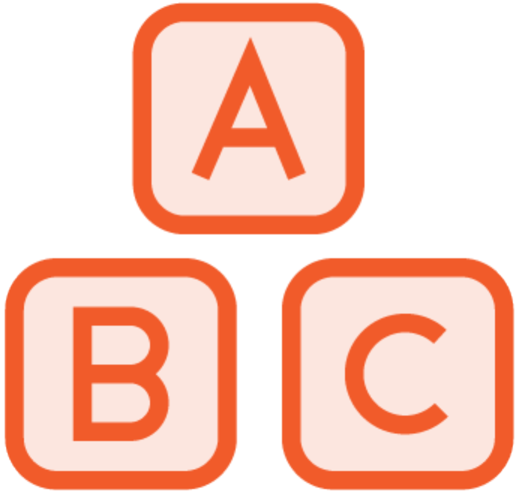
What this course will cover

What Is Microsoft SQL Server?

SQL Server is Microsoft’s relational database management system. The primary function of the software system is storing and retrieving data requested by other applications.

SQL Server Versions and Editions

SQL Server comes in numerous versions – based on year of release since SQL Server 2000



Various editions exist based upon features, functionality, and price

SQL Server 2012/2014

Developer

Express

Web

Standard

Enterprise

Business Intelligence

SQL Server 2016

Developer

Express

Web

Standard

Enterprise

SQL Server 2017

- Developer
- Express
- Standard
- Enterprise
- Linux

COURSE OBJECTIVE

Provide an understanding of SQL Server database fundamentals

Introduce SQL Server in general

Explain various relational database concepts

Discuss how to create different database objects

COURSE OBJECTIVE

Provide an understanding of SQL Server database fundamentals

Show how to manipulate data

Discuss reasons for normalization and data storage

Explain administration techniques to secure a database and perform backups

Course Focus and Structure

Understanding Core Database Concepts

How to Create Database Objects

Manipulating Data

Understanding Data Storage

How to Administer a Database

Terminology used in this course

This course is designed to help you better understand database fundamentals while preparing you for the certification exam 98-364. Some terms used in this course align with the exam, however they may have different connotations outside of the exam.

What We Covered

Introduction to SQL Server

Versions and editions of SQL Server

What this course will cover

Understanding Core Database Concepts

Tim Radney

PRINCIPAL CONSULTANT

@tradney www.SQLskills.com

Module Overview

Relational database concepts

How data is stored in tables

Data Manipulation Language (DML)

Data Definition Language (DDL)

What Is a Database?

A database is an organized collection of data, typically stored and accessed electronically from a computer system or electronic device.

Types of Databases

Hierarchical

(Directory structure)

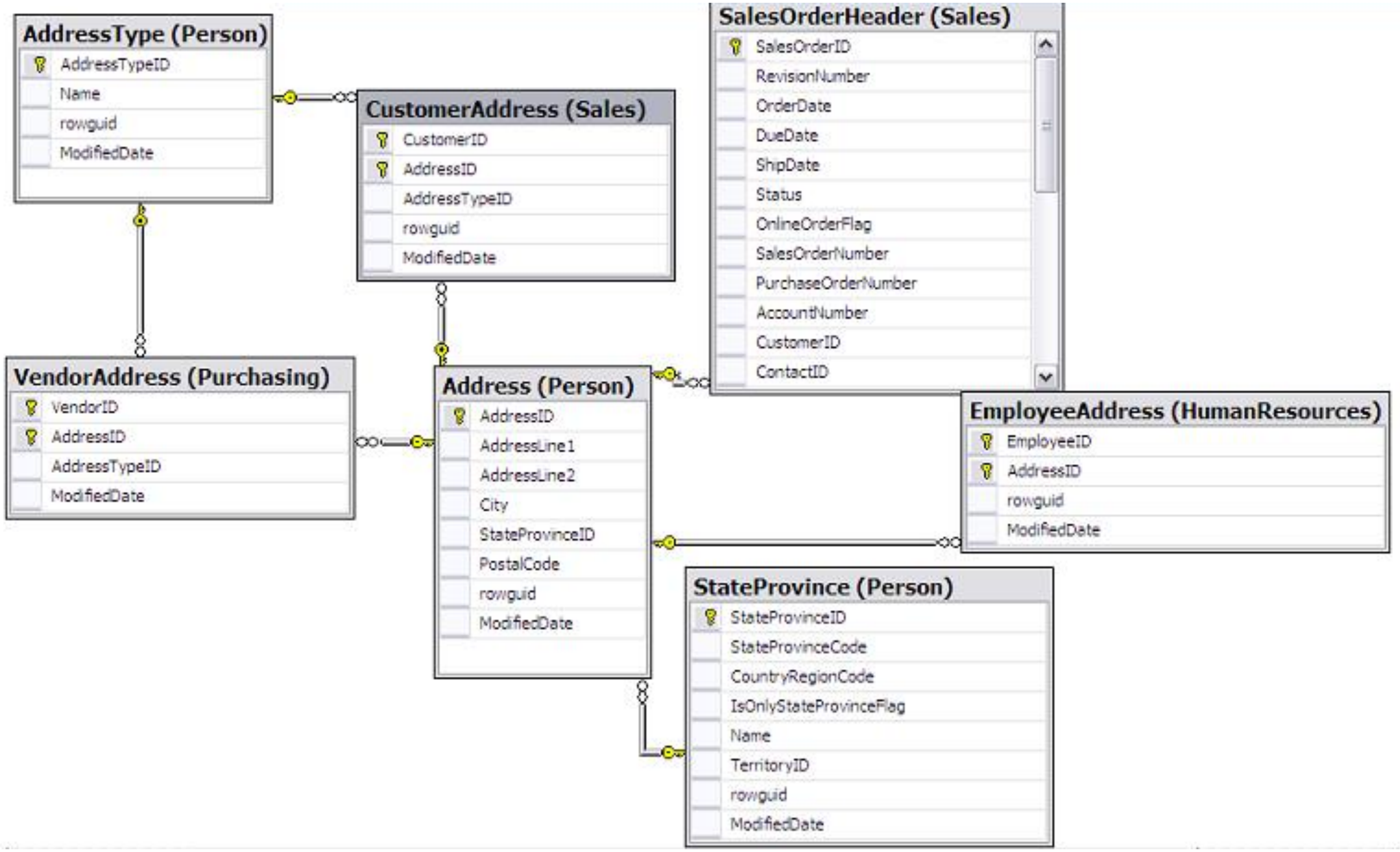
Flat-file

(CSV, Excel, delimited)

Relational

(SQL Server, Oracle, MySQL, and more)

Relational Database



AddressType (Person)
- AddressTypeID
- Name
- rowguid
- ModifiedDate

CustomerAddress (Sales)
- CustomerID
- AddressID
- AddressTypeID
- rowguid
- ModifiedDate

SalesOrderHeader (Sales)
- SalesOrderID
- RevisionNumber
- OrderDate
- DueDate
- ShipDate
- Status
- OnlineOrderFlag
- SalesOrderNumber
- PurchaseOrderNumber
- AccountNumber
- CustomerID
- ContactID

VendorAddress (Purchasing)
- VendorID
- AddressID
- AddressTypeID
- ModifiedDate

Address (Person)
- AddressID
- AddressLine1
- AddressLine2
- City
- StateProvinceID
- PostalCode
- rowguid
- ModifiedDate

EmployeeAddress (HumanResources)
- EmployeeID
- AddressID
- rowguid
- ModifiedDate

StateProvince (Person)
- StateProvinceID
- StateProvinceCode
- CountryRegionCode
- IsOnlyStateProvinceFlag
- Name
- TerritoryID
- rowguid
- ModifiedDate

Relational Database Concepts

Table

(Think of an Excel sheet)

Column

(Think of a column in an Excel sheet)

Row

(Think of a row in an Excel sheet)

Data Manipulation Language (DML)

- SELECT
- INSERT
- UPDATE
- DELETE
- MERGE
- BULK INSERT

Data Manipulation Language

Statement	Description
SELECT	Retrieves rows from database and enables selection of one or many rows or columns from one or many tables
INSERT	Adds one or more rows to a table or view
UPDATE	Changes existing data in a table or view
DELETE	Removes one or more rows from a table or view
MERGE	Performs insert, update, or deletes on a target table based on the results of a join with a source table
BULK INSERT	Imports a data file into a database table or view

Data Manipulation Language

- WRITETEXT
- READTEXT
- UPDATETEXT

Data Manipulation Language

Statement	Description
WRITETEXT	Permits minimally logged, interactive updating of an existing text, ntext, or image column by overwriting any existing data (does not work on views)
READTEXT	Reads text, ntext, or image values from a text, ntext, or image column by starting from a specified offset and reading a specified number of bytes
UPDATETEXT	Updates an existing text, ntext, or image field or a part of a text, ntext, or image column in place

Data Definition Language (DDL)

CREATE

ALTER

DROP

Collations

Data Definition Language

Statement	Description
CREATE	Enables creating a item in SQL Server. For example, database, table, view, users, index, and more
ALTER	Enables modifying an existing object. For example, database, table, view, and more
DROP	Drops an existing object
Collations	Defines the collation of a database or table column, or a collation cast operation when applied to a character string expression

Data Definition Language

ENABLE TRIGGER

DISABLE TRIGGER

RENAME

UPDATE STATISTICS

Data Definition Language

Statement	Description
ENABLE TRIGGER	Enables a DML, DDL, or login trigger
DISABLE TRIGGER	Disables a trigger
RENAME	Renames a user-created table in SQL Data Warehouse or a user-create table or database in Parallel Data Warehouse
UPDATE STATISTICS	Updates query optimization statistics on a table or indexed view.

Additional Common DDL Commands

Statement	Description
USE	Changes the database context to the specified database or database snapshot in SQL Server Does not apply to Azure SQL Database or Azure SQL Data Warehouse
TRUNCATE	Used to remove all rows from a table or specified partitions of a table without logging the individual row deletions

Demo

Tables and columns

What We Covered

Relational database concepts

How data is stored in tables

Data Manipulation Language (DML)

Data Definition Language (DDL)

How to Create Database Objects

INTRODUCTION

Tim Radney
PRINCIPAL CONSULTANT

@tradney www.SQLskills.com

Module Overview

Data types

Tables and views

Stored procedures and functions

What Are Data Types?

An attribute that specifies the type of data an object can hold, such as date and time, binary strings, monetary data, integer data.

Data Type Characteristics

Each column, local variable, expression, and parameter has a related data type

Data type of result is determined by applying rules of data type precedence to data types of input expressions

Collation of result is determined by rules of collation precedence when result data type is char, varchar, text, nchar, nvarchar, ntext

Precision, scale, and length of result depend on precision, scale, and length of input expressions

Data Types

Exact numerics

Approximate numerics

Unicode character strings

Binary strings

Exact Numerics

Statement	Description
tinyint, smallint, int, bigint	Exact-number data types that use integer data Always use the smallest data type for your purpose (tinyint : 0 to 255 : 1 byte) (smallint : -2^15 to 2^15 : 2 bytes) (int : -2^31 to 2^32 : 4 bytes) (bigint : -2^63 to 2^63 : 8 bytes)
decimal, numeric	Fixed precision and scale, storing -10^38 + 1 to 10^38 – 1 Scale is number of decimals to right of decimal point (1 to 38 bytes with a default of 18)
bit	An integer that can take the value of 0, 1, or NULL
smallmoney, money	Accurate to a ten-thousandth Use a period to separate monetary units such as cents (smallmoney : 4 bytes) (money : 8 bytes)

Approximate Numerics

Statement	Description
float	Approximate-number data type to store the mantissa of the float number in scientific notation and dictates the precision and storage size (float [(n)] : n = 1 to 24 : 7 digits and 4 bytes) (float [(n)] : n = 25 to 53 : 15 digits and 8 bytes) Default value for n is 53
real	7 digits of precision making it identical to float(24) using 4 bytes

Unicode Character Strings

Statement	Description
nchar	Fixed-length string data nchar[(n)] n defines the string length in byte-pairs and n = 1 to 4,000 (n x 2 bytes)
nvarchar	Variable-length string data nvarchar[(n)] defines the string length in byte-pairs and n = 1 to 4,000 (n x 2 bytes)
ntext	Variable-length Unicode data with n = 0 to 2^30 – 1 (n x 2 bytes)

Binary Strings

Statement	Description
binary	Fixed-length binary data with a length of n = 1 to 8,000 (n bytes)
image	Variable-length binary data of 0 to 2^31 - 1 bytes (2GB)
varbinary	Variable-length binary data varbinary [(n)] has a length of n = 1 to 8,000 varbinary [(max)] has a length of n = 1 to 2^31 - 1 (2GB) Storage is the actual length of the data + 2 bytes

More Data Types

Date and time

Character strings

Miscellaneous data types

Date and Time

Statement	Description
date	Defines a date in SQL Server
datetime	Date with time of day with fractional seconds and based on 24-hour clock
datetime2	Expanded datetime with larger date range, default fractional precision, and user-specified precision
datetimeoffset	Date with time of day that has time zone awareness and based on 24-hour clock
smalldatetime	Datetime without seconds or fractional seconds
time	Time of a day without time zone awareness

Character Strings

Statement	Description
char	Fixed-length string data char [(*n*)] has a length of n = 1 to 8,000 bytes
varchar	Variable-length string data varchar [(n)] has a length of n = 1 to 8,000 varchar [(max)] has a length of n = 1 to 2^31 - 1 (2GB) Storage is the actual length of the data in bytes
text	Variable-length non-Unicode data in the code page of the server and a maximum length of 2^31 - 1 (2GB)

Miscellaneous Data Types

Statement	Description
cursor	For variables or stored procedure OUTPUT parameters that contain a reference to a cursor
rowversion	Exposes automatically generated, unique binary numbers within a database that are generally used as a mechanism for version-stamping table rows
hierarchyid	Variable length, system data type that represents a position in a tree hierarchy
uniqueidentifier	6-byte GUID created from NEWID or NEWSEQUENTIALID functions or converting a string constant in a certain format

Miscellaneous Data Types

Statement	Description
sql_variant	Stores values of various SQL Server-supported data types
xml	Data type that stores XML data
spatial geometry and geography types	Implemented as a .NET common language runtime (CLR) data type
table	Primarily used for temporary storage of a set of rows returned as the result set of a table-valued function

Demo

Data types

Tables

Contains all the data in the database in a row-and-column format

A table can have 1,024 columns, and 30,000 columns if using SPARSE

Assign properties to a table and columns and use compression

Views

Virtual table defined by a query

Similar to a table with named columns and rows that are produced dynamically when referenced

Acts like filter on tables and can reference multiple tables and provide a level of security

Demo

Tables and views

Stored Procedures

A group of one or more Transact-SQL statements or a reference to a Microsoft .Net Framework CLR method

Can accept input parameters and return multiple values as output parameters

Have programming statements that perform operations within the database

Return a status value indicating success or failure

Stored Procedure Benefits

Reduced server/client network traffic

Stronger security

Reuse of code

Easier maintenance

Improved performance

Functions

Functions are routines that accept parameters, perform an action, and return the result of that action as a value

Functions

Allow modular programming

Allow for faster execution

Can reduce network traffic

Scalar function

Table-valued functions

System functions

Demo

Stored procedures and functions

What We Covered

Data types

Tables and views

Stored procedures and functions

Manipulating Data

INTRODUCTION

Tim Radney
PRINCIPAL CONSULTANT

@tradney www.SQLskills.com

Module Overview

Selecting data

Updating data

Deleting data

Inserting data

Selecting Data

Statement	Description
SELECT	Retrieves rows from the database and enables you to select one or more columns or rows from one or many tables
FROM	Specifies the tables, views, derived tables, and joined tables used in DELETE, SELECT, and UPDATE statements
GROUP BY	Clause that divides the query result into groups of rows, typically for performing aggregations
HAVING	Typically used with GROUP BY in a SELECT statement as a search condition for a group or an aggregate

Selecting Data

Statement	Description
ORDER BY	Orders result set of a query by the specified column list
UNION	Combines results of two or more queries into a single result set that has all rows of each query in the union
EXCEPT	Returns distinct rows from left input query that aren’t output by right input query
INTERSECT	Returns distinct rows that are output by both left and right input queries
JOIN	Defines the way two tables are related in a query

Demo

Various SELECT statements

```
UPDATE table_name
SET column = value
WHERE [condition];

UPDATE table
SET column1 = value, column2 = value;
```

UPDATE Statement

Changes existing data in a table or view

UPDATE statements always acquires an exclusive (X) lock on the row(s) it modifies

Demo

Using the UPDATE statement

```
DELETE FROM table_name
WHERE [condition];
```

DELETE statement

Used to delete existing records in a table

DELETE permissions are required on the target table as well as SELECT permissions if the statement contains a WHERE clause (and if no WHERE clause is used, all records will be deleted)

```
TRUNCATE table_name;
```

TRUNCATE statement

Removes all rows from a table or specified partitions of a table but doesn’t log row deletions so much faster than DELETE if removing all rows in a table from using fewer system and transaction log resources

Minimum permission required is ALTER on table_name

Demo

Using the DELETE and TRUNCATE statements

```
INSERT INTO table_name (column1, column2, column3)
VALUES (value1, value2, value3);

INSERT INTO table_name
VALUES (value1, value2, value3);
```

INSERT statement

Allows you to insert new records into a table

If adding values for all columns in table, no need to specify the column names, otherwise order of values must match order of columns

INSERT permission is required

Demo

Using the INSERT statement

What We Covered

Selecting data

Updating data

Deleting data

Inserting data

Understanding Data Storage

INTRODUCTION

Tim Radney
PRINCIPAL CONSULTANT

@tradney www.SQLskills.com

Module Overview

Normalization

Primary keys

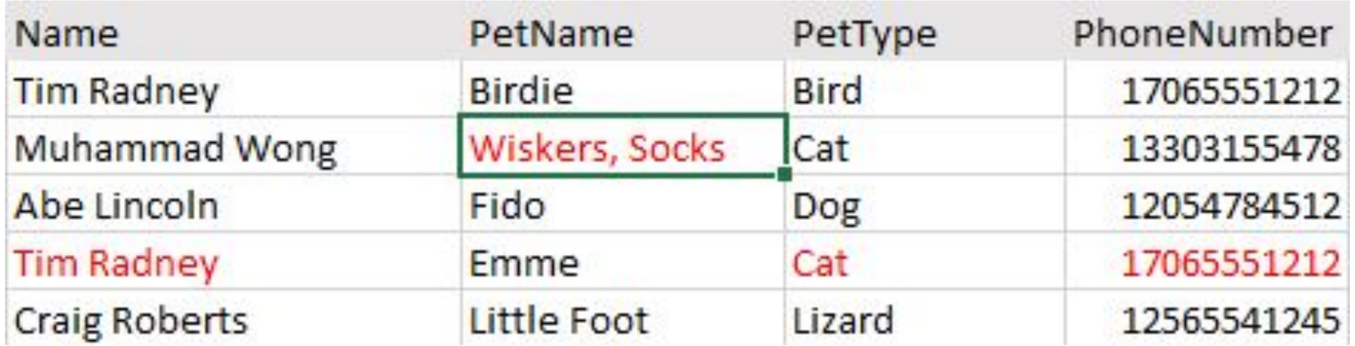
Foreign keys

Composite keys

Normalization

Resolving issues of data redundancy and improving storage efficiency, data integrity, scalability

Helps reduce insert, update, and delete anomalies



Name	PetName	PetType	PhoneNumber
Tim Radney	Birdie	Bird	17065551212
Muhammad Wong	Wiskers, Socks	Cat	13303155478
Abe Lincoln	Fido	Dog	12054784512
Tim Radney	Emme	Cat	17065551212
Craig Roberts	Little Foot	Lizard	12565541245

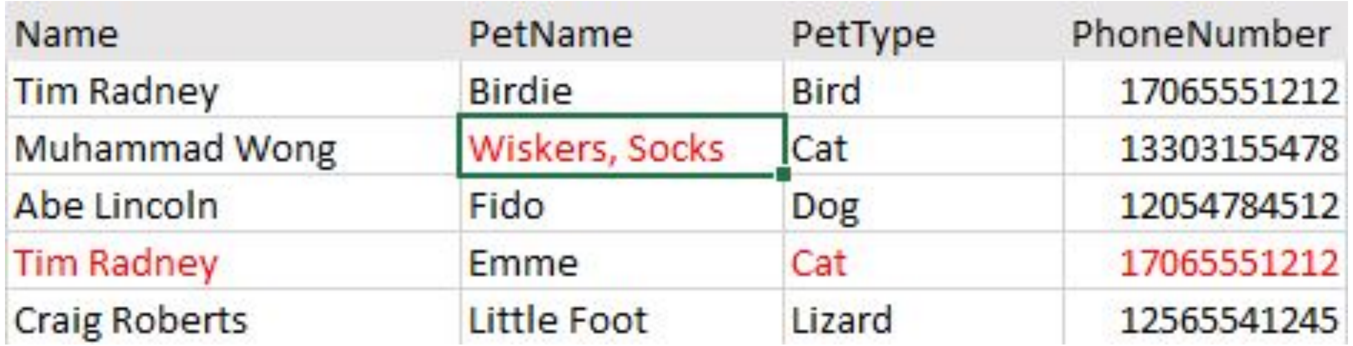
Normalization

Statement	Description
First Normal Form (1NF)	Eliminates repeating groups. No duplicate records exist in table and no multi-valued attributes. All entries in column are of same data type
Second Normal Form (2NF)	Table is already in 1NF, all partial dependencies are removed to another table(s)
Third Normal Form (3NF)	Table is already in 2NF, eliminate non-dependent columns and transient dependencies.
Fourth Normal Form (4NF)	Already in 3NF, no independent multiple relationships
Fifth Normal Form (5NF)	Already in 4NF, isolate semantically-related relationships

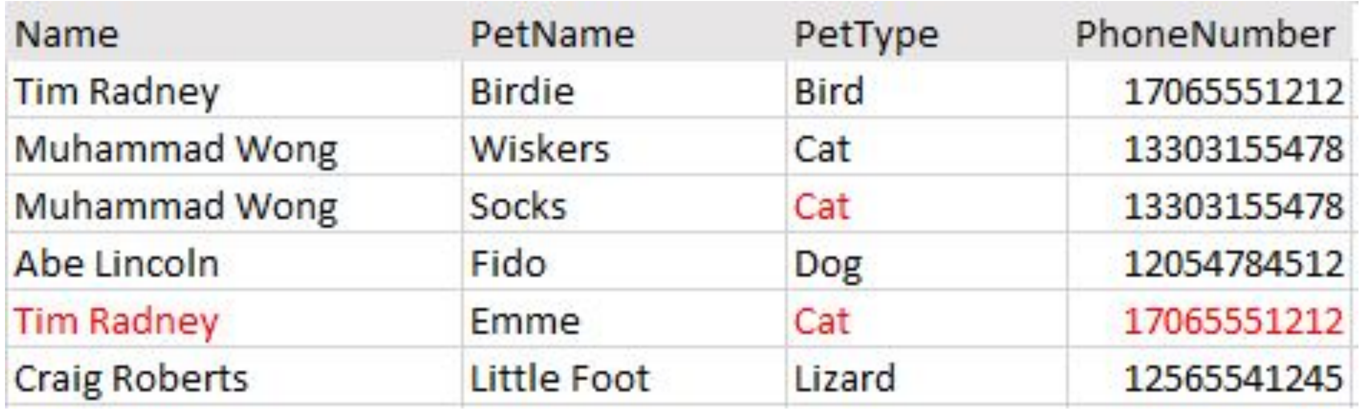
First Normal Form

Top image violates 1NF due to attributes not being single

We can fix this by splitting the attributes into another row



Name	PetName	PetType	PhoneNumber
Tim Radney	Birdie	Bird	17065551212
Muhammad Wong	Wiskers, Socks	Cat	13303155478
Abe Lincoln	Fido	Dog	12054784512
Tim Radney	Emme	Cat	17065551212
Craig Roberts	Little Foot	Lizard	12565541245

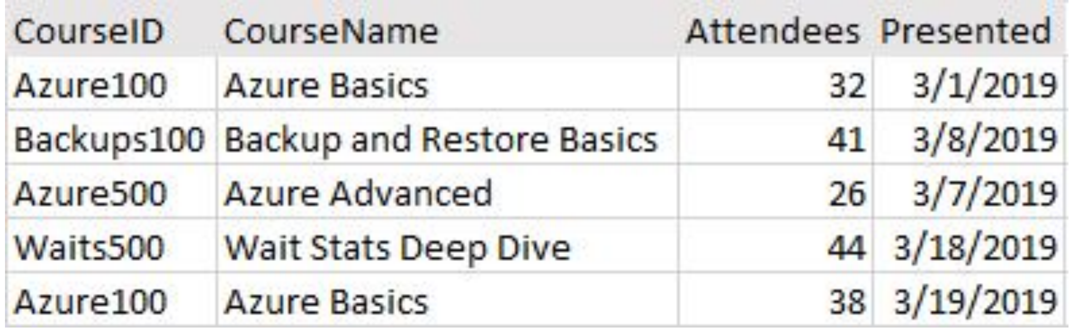


Name	PetName	PetType	PhoneNumber
Tim Radney	Birdie	Bird	17065551212
Muhammad Wong	Wiskers	Cat	13303155478
Muhammad Wong	Socks	Cat	13303155478
Abe Lincoln	Fido	Dog	12054784512
Tim Radney	Emme	Cat	17065551212
Craig Roberts	Little Foot	Lizard	12565541245

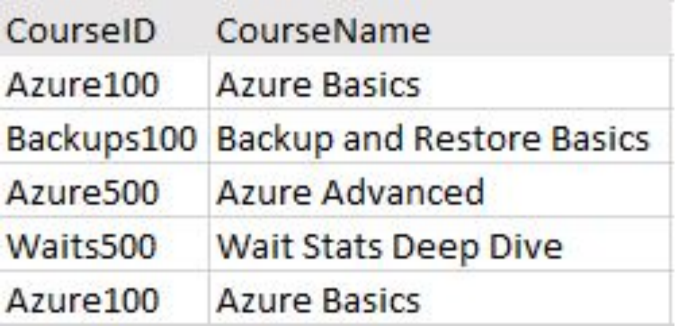
Second Normal Form

Top image violates 2NF as CourseName depends on CourseID

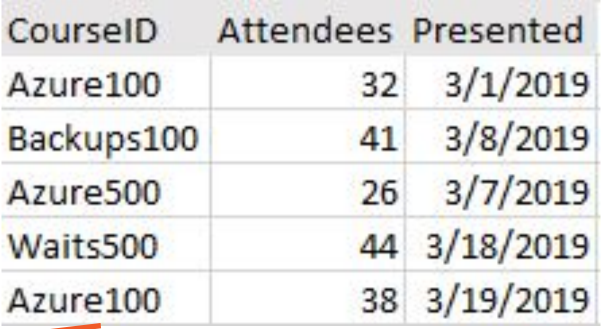
CourseID and CourseName should create a new table



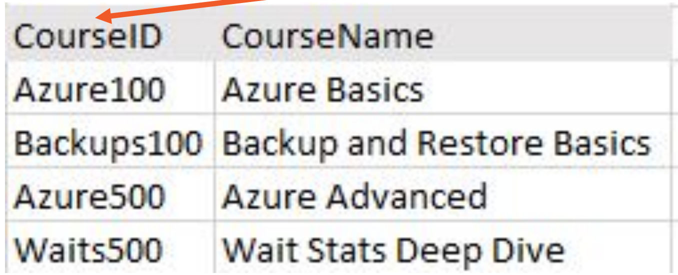
CourseID	CourseName	Attendees	Presented
Azure100	Azure Basics	32	3/1/2019
Backups100	Backup and Restore Basics	41	3/8/2019
Azure500	Azure Advanced	26	3/7/2019
Waits500	Wait Stats Deep Dive	44	3/18/2019
Azure100	Azure Basics	38	3/19/2019



CourseID	CourseName
Azure100	Azure Basics
Backups100	Backup and Restore Basics
Azure500	Azure Advanced
Waits500	Wait Stats Deep Dive
Azure100	Azure Basics



CourseID	Attendees	Presented
Azure100	32	3/1/2019
Backups100	41	3/8/2019
Azure500	26	3/7/2019
Waits500	44	3/18/2019
Azure100	38	3/19/2019

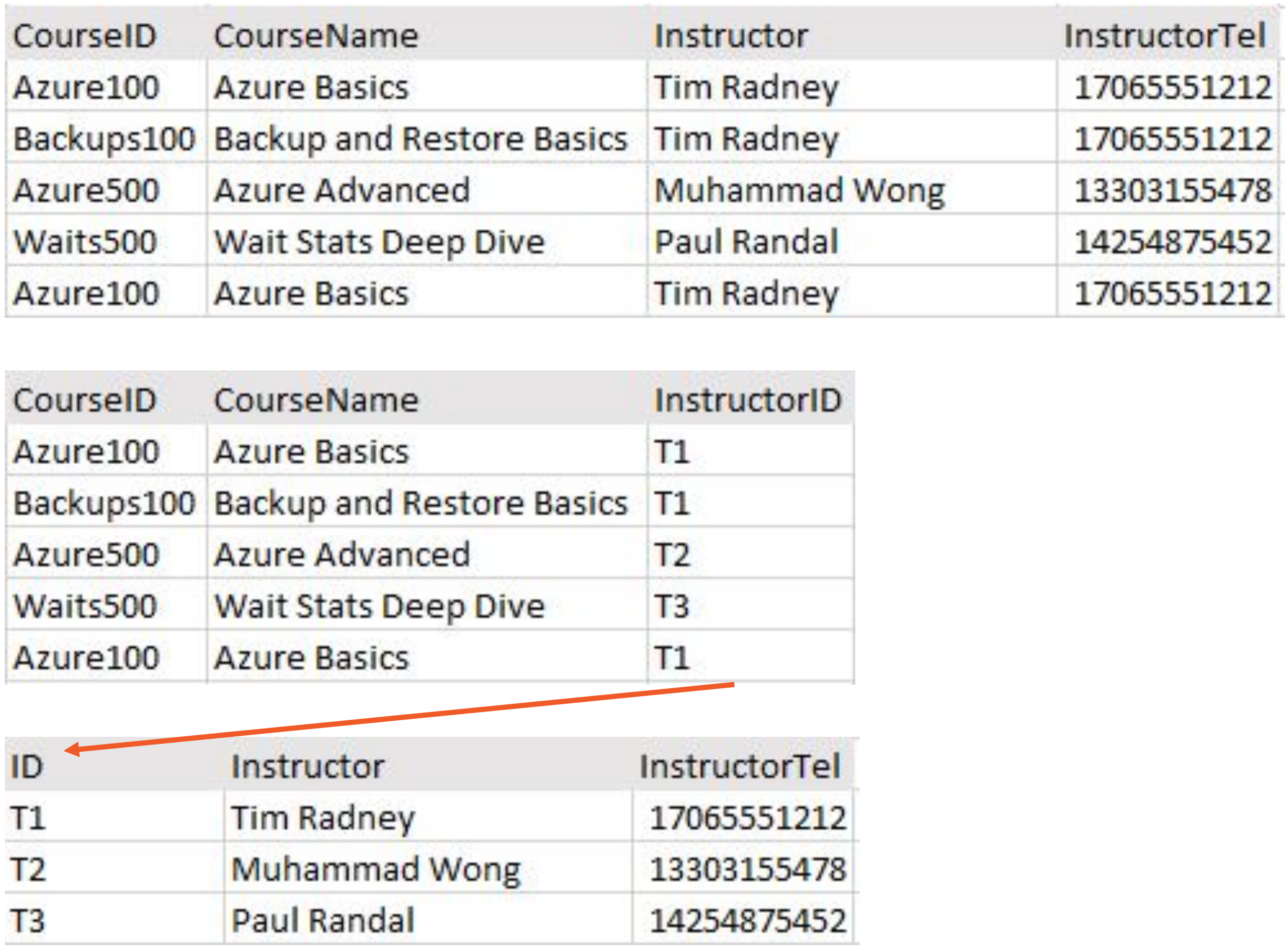


CourseID	CourseName
Azure100	Azure Basics
Backups100	Backup and Restore Basics
Azure500	Azure Advanced
Waits500	Wait Stats Deep Dive

Third Normal Form

The top image violates 3NF due to InstructorTel being a non-key attribute.

InstructorTel has a transient dependency on Instructor



CourseID	CourseName	Instructor	InstructorTel
Azure100	Azure Basics	Tim Radney	17065551212
Backups100	Backup and Restore Basics	Tim Radney	17065551212
Azure500	Azure Advanced	Muhammad Wong	13303155478
Waits500	Wait Stats Deep Dive	Paul Randal	14254875452
Azure100	Azure Basics	Tim Radney	17065551212

CourseID	CourseName	InstructorID
Azure100	Azure Basics	T1
Backups100	Backup and Restore Basics	T1
Azure500	Azure Advanced	T2
Waits500	Wait Stats Deep Dive	T3
Azure100	Azure Basics	T1

ID	Instructor	InstructorTel
T1	Tim Radney	17065551212
T2	Muhammad Wong	13303155478
T3	Paul Randal	14254875452

Primary Key

Uniquely identifies each record in a table

Cannot contain NULL values, must be UNIQUE

Only one primary key per table, using single or multiple fields

Foreign Key

Not just to a primary key constraint in another table, can reference columns of a UNIQUE constraint in another table

Can only reference tables within the same database on the same server

Self-reference is allowed to reference another column in the same table

Column-level constraint must list only one reference column and be same data type

Table-level constraint needs same number of reference columns and data types

Composite Key

A key that includes multiple columns. Can be designated as the primary key if the combination of the columns result in a unique composite value for every row in the table. Commonly used in tables designed with many to many relationships.

Demo

Primary, foreign, and composite keys

What We Covered

Normalization

Primary keys

Foreign keys

Composite keys

How to Administer a Database

INTRODUCTION

Tim Radney
PRINCIPAL CONSULTANT

@tradney www.SQLskills.com

Module Overview

Authentication

Permissions

Maintenance considerations

Backup and restore

Authentication

Windows Authentication

Commonly referred to as Integrated Security

SQL Server and Windows Authentication

Commonly referred to as Mixed-mode Security

Authentication

Statement	Description
Windows logins	Provides single sign-on for the OS and SQL Server Windows Group, Domain User Account, Local User Account
SQL logins	Allows application-specific security and mixes OS access SQL Server Login – Server Role Database User – Role, Application Role, or Group

Sysadmin

“Superuser role”

SA is mapped to sysadmin

SA is very important and must be secured

Database Users

Windows logins

SQL logins

Windows groups

User without login

Special Users

Statement	Description
DBO	Database Owner, the (one) owner of the database
	Sysadmin mapped to DBO in every database
	DBO bypasses security checks on database objects
Guest	Logins without a mapping, map to Guest
	Guest cannot connect by default to user databases

Database Permissions

GRANT

Positive privilege

DENY

Negative privilege that overrides GRANT

REVOKE

Negate a GRANT or DENY

Database Permissions

ALTER
BACKUP
CONNECT
CONTROL
CREATE
DELETE
DROP
EXECUTE
INSERT
TAKE OWNERSHIP
SELECT
UPDATE
VIEW DEFINITION

Objects and Permissions

Instance-level

ENDPOINTs

Database-level

ASSEMBLY and SERVICEs

Schema-level

TABLEs, VIEWs, PROCEDUREs, QUEUEs

Column-level

Database Roles

Statement	Description
db_owner	Perform all configuration and maintenance, including dropping the database
db_securityadmin	Modify role membership and manage permissions
db_accessadmin	Add or remove access to the database for Windows logins or groups, and SQL Server logins
db_backupoperator	Can back up the database
db_ddladmin	Can run any DDL command in a database
db_datawriter	Can add, delete, or change data in all user tables

Database Roles

Statement	Description
db_datareader	Can read all data from all user tables
db_denydatawriter	Cannot add, modify, or delete any data in the user tables within a database
db_denydatareader	Cannot read any data in the user tables within a database

Additional notes

Public role: equivalent to the everyone role in Windows, all users are permanent members of the role

Each database contains its own users

You can create your own roles

A user can be a member of 1-N roles

Demo

Users, roles, and permissions

Maintenance Considerations

Statistics

Indexes

Consistency checks

Statistics

Used by the Query Optimizer to compile an optimal (or ‘good enough’) execution plan

Out-of-date statistics can negatively impact the Query Optimizer’s plan choice

“Auto Update Statistics” – typically updates after about 20% + 500 rows change

You need a process to programmatically update statistics to not rely on automatic updates

Indexes

Fragmentation naturally occurs due to inserts, updates, and deletes

Fragmentation can negatively impact performance

Controlling fragmentation:

- Rebuild, reorganize, or disable-and-rebuild the index
- Database Maintenance Plans
- Custom Scripts in a SQL Agent Job
- Third-party tools

Consistency Checks

Corribution happens

- Generally due to I/O subsystem issue

Finding corruption

- DBCC CHECKDB
- DBCC CHECKALLOC
- DBCC CHECKCATALOG
- DBCC CHECKFILEGROUP
- DBCC CHECKTABLE

You need to have a schedule job to run DBCC CHECKDB often enough to prevent potential data loss

Demo

Maintenance processes

Backup Types

- Full
- Differential
- Log
- Filegroup
- File
- COPY_ONLY

What Is a Tail-log Backup?

Captures any log records that have not yet been backed up to prevent data loss and to keep the log chain intact

Recovering Transactions

Do you need to perform a tail-log backup?

Restore the last known-good full backup WITH NORECOVERY

Restore the most recent differential backup WITH NORECOVERY, if you have them

Restore transaction logs WITH NORECOVERY in order

Restore any tail-log backup

Finally, RESTORE dbname WITH RECOVERY

Have a plan

Know how to recover

- Test that your backups are good
- Have a disaster recovery plan
- Test your disaster recovery plans
- Know how long restores take
- Plan a recovery strategy, not a backup strategy

Demo

Backup, restore, and recovery models

What We Covered

Authentication

Permissions

Maintenance considerations

Backup and restore

What now?

Start Exploring SQL Server

Download and install SQL Server Developer Edition

Step through all the demos and watch modules that you are less familiar with

Schedule to take the Understanding Database Fundamentals Exam 98-364

Watch more courses

SQL Server: Transact-SQL Basic Data Modification

SQL Server: Transact-SQL Basic Data Retrieval

SQL Server: Why Physical Database Design Matters

SQL Server: Indexing for Performance

Course Summary

Provided an initial understanding of SQL Server and core database concepts

Covered how to create database objects and manipulate data

Discussed how data storage is handled and demonstrated how to administer a database