



DATABASE DESIGN

Julie Allen 2024



DATABASE

INTEGRITY

RECAP: NORMALISATION

1NF Conditions:

1. There are no repeating groups
2. All fields are unique
3. All values are atomic
4. Each table has a primary key

2NF Conditions:

1. Tables are first in 1NF
2. All non-key attributes are dependent on ALL parts of the primary key

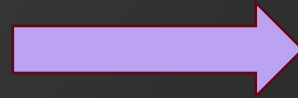
3NF Conditions:

1. Tables are first in 2NF
2. No non-key attributes are dependent on other non-key attributes.

DOCTOR-APPOINTMENT SYSTEM: 1NF

Doctor Appointment System - ONF

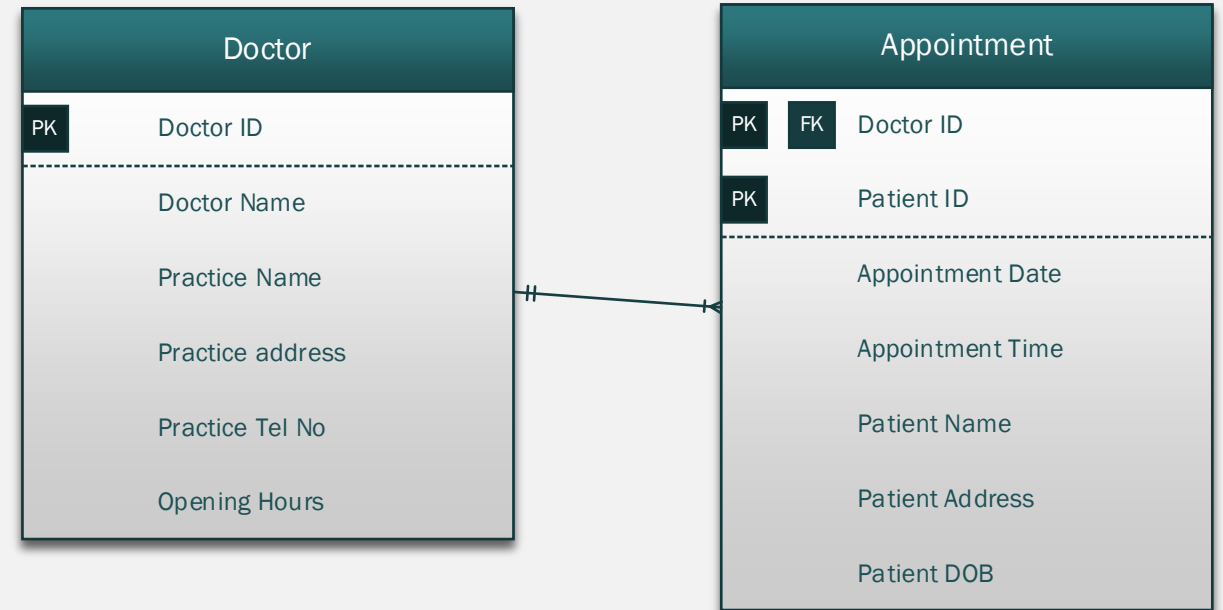
PK	Doctor ID
	Doctor Name
	Practice Name
	Practice address
	Practice Tel No
	Opening Hours
	Appointment Date
	Appointment Time
	Patient ID
	Patient Name
	Patient Address
	Patient DOB



The repeating groups are the Patient Appointments – one doctor will have many patient appointments.

We take a copy of the primary key (DoctorID) to create the link between the tables, which becomes a foreign key in the new table.

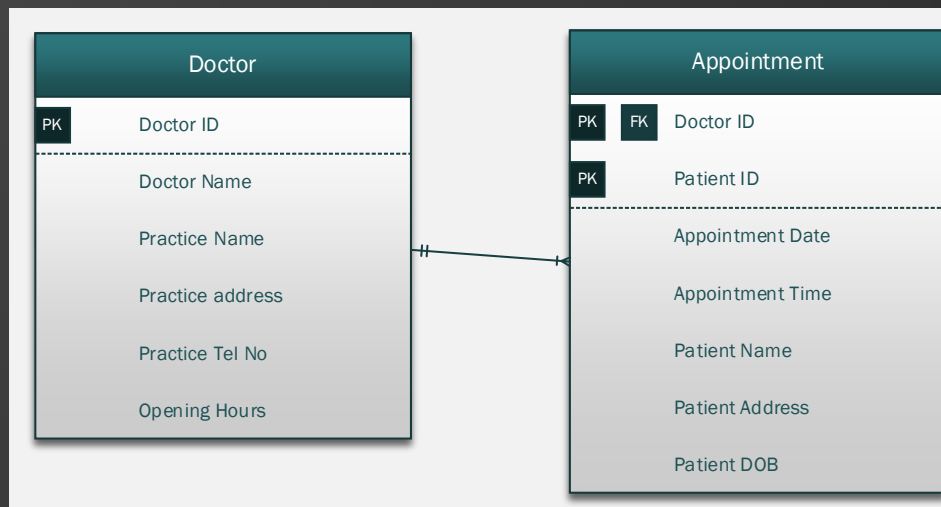
Take out the repeating groups



The primary key in our new table is a composite key made up of DoctorID and PatientID.

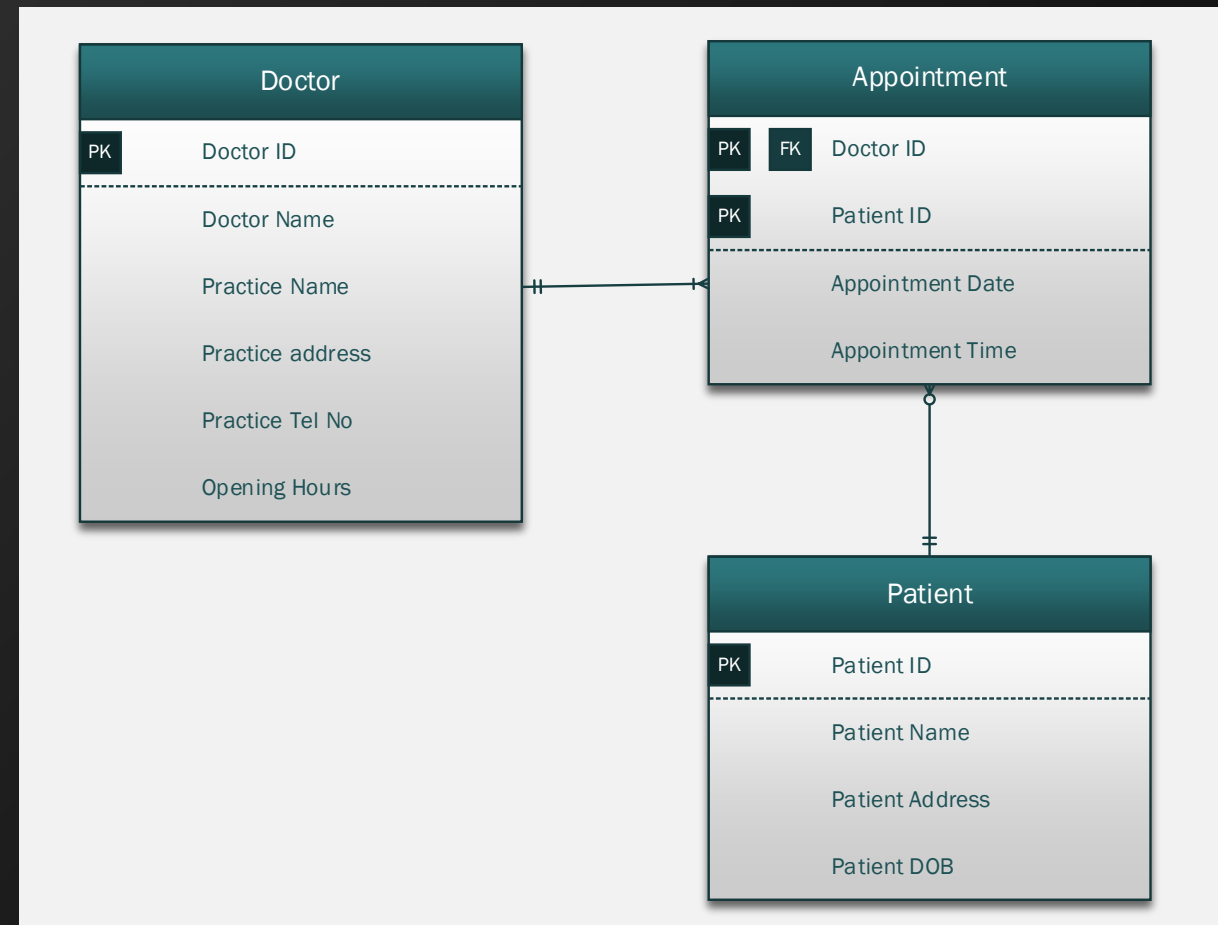
DOCTOR-APPOINTMENT SYSTEM: 2NF

All non-key attributes must be dependent on ALL parts of the primary key



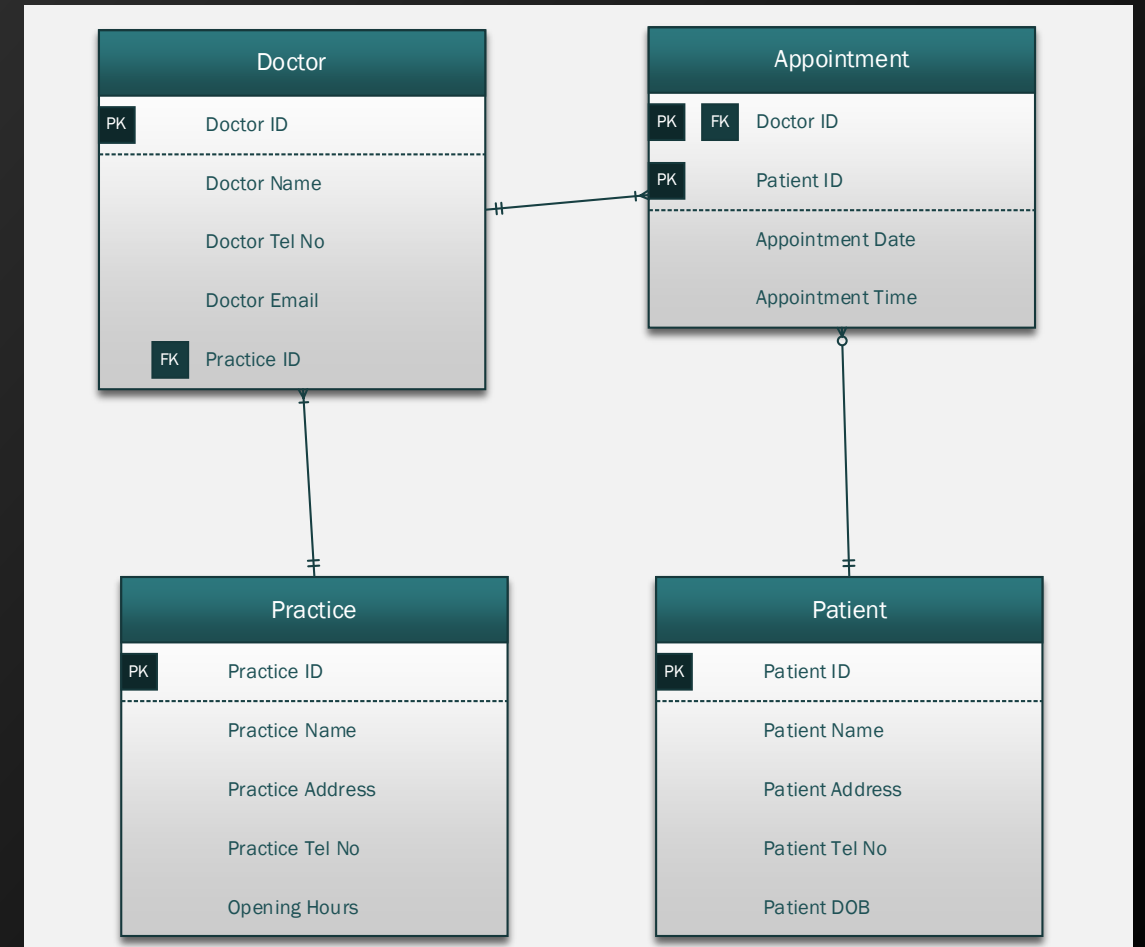
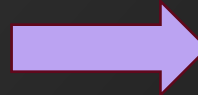
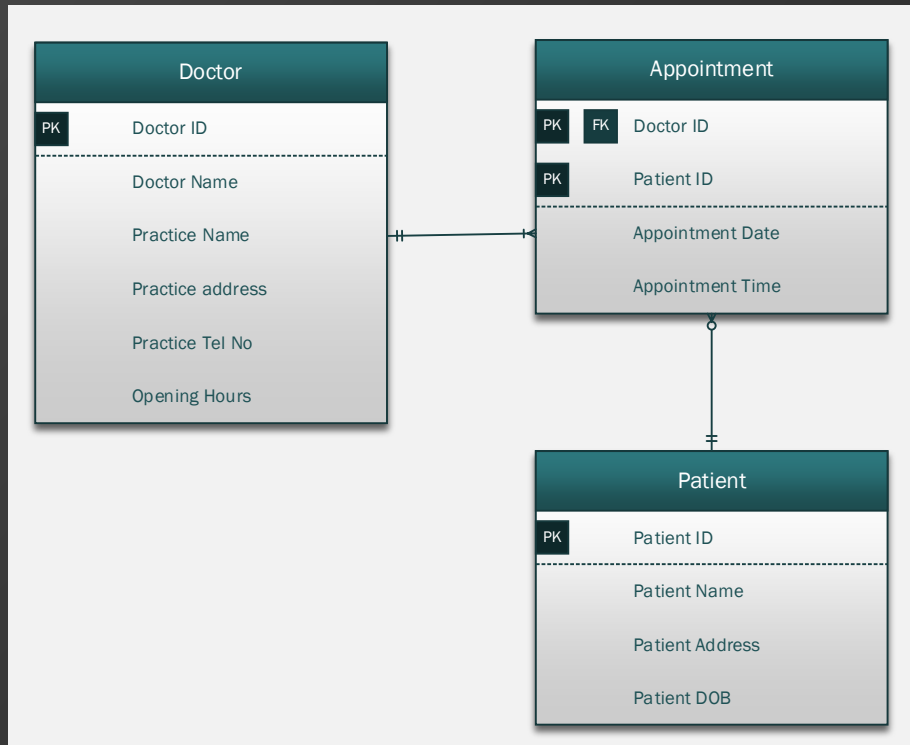
The Doctor table only has one attribute making up its primary key and as such is already in 2NF.

In the Appointment table the patient details are dependent ONLY on the PatientID, as such we take them out and create a new table called Patient.



DOCTOR-APPOINTMENT SYSTEM: 3NF

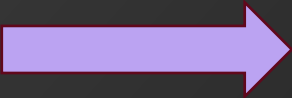
No non-key attributes are dependent on other non-key attributes.



In the Doctor table the Practice details are dependent on each other, so we take them out and create a new table called Practice with a PracticeID attribute for the primary key.

College	
PK	BatchNumber

	TutorCode
	TutorFirstName
	TutorSurname
	TutorAddressLine1
	(more tutor details)
	AssignmentsMarked
	DateSent
	DateReceived
	StudentCode
	StudentFirstName
	StudentSurname
	(other student details)
	CourseCode
	CourseName
	(other course details)
	AssignmentNo

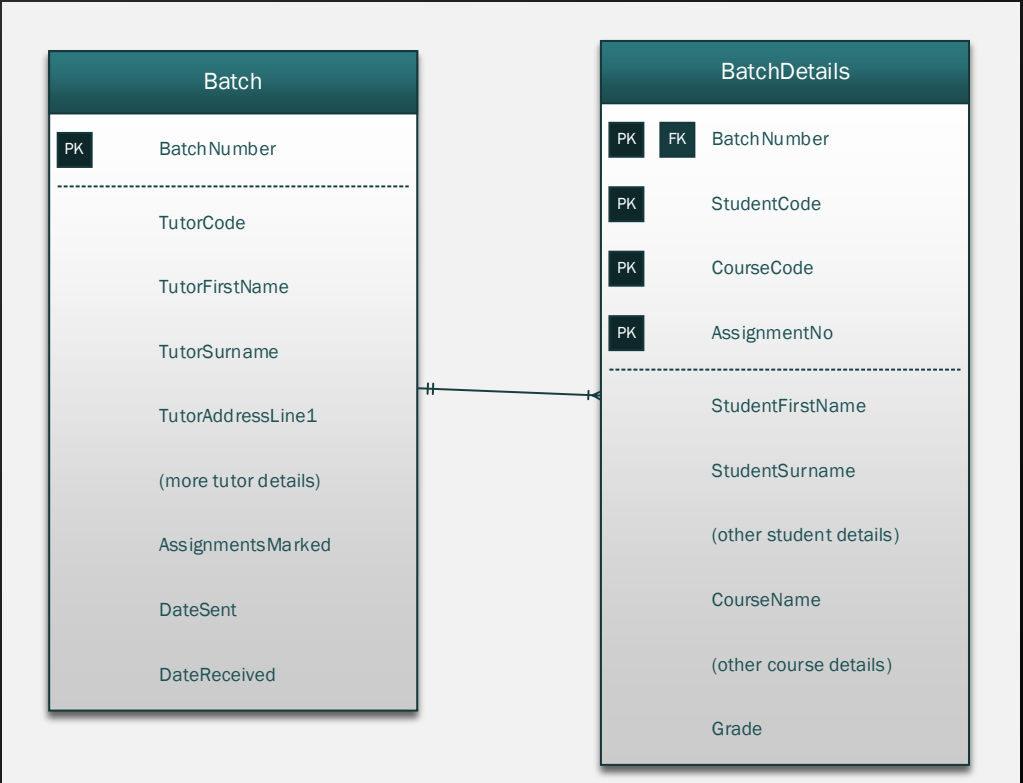


The repeating groups are the student assignments – one batch will have many student assignments (batch details).

We take a copy of the primary key (Batch Number) to create the link between the tables, which becomes a foreign key in the new table.

COLLEGE BATCH MARKING SYSTEM: 1NF

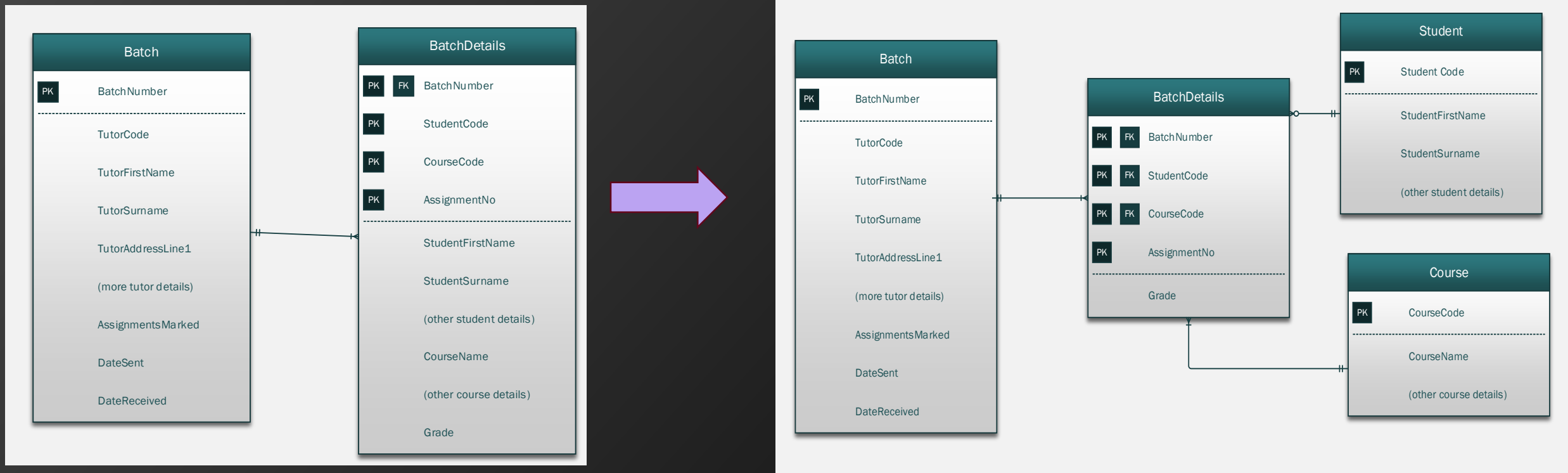
Take out the repeating groups



The primary key in our new table is a composite key made up of BatchNumber, StudentCode, CourseCode, and AssignmentNo.

COLLEGE BATCH MARKING SYSTEM: 2NF

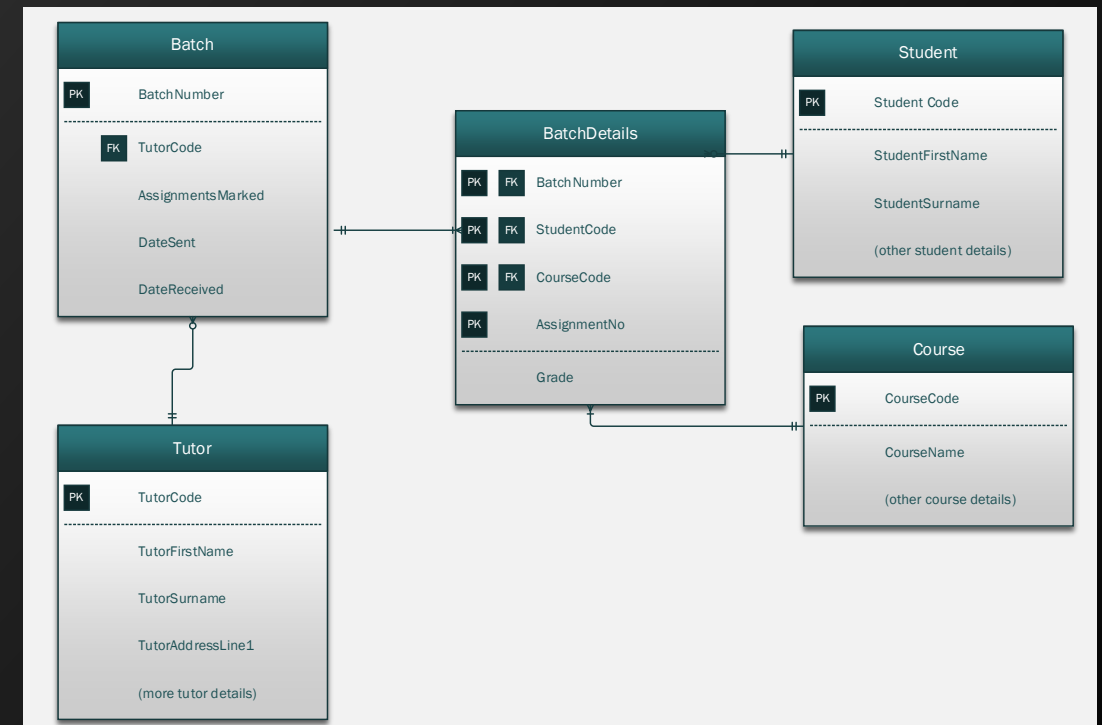
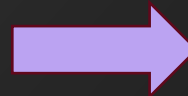
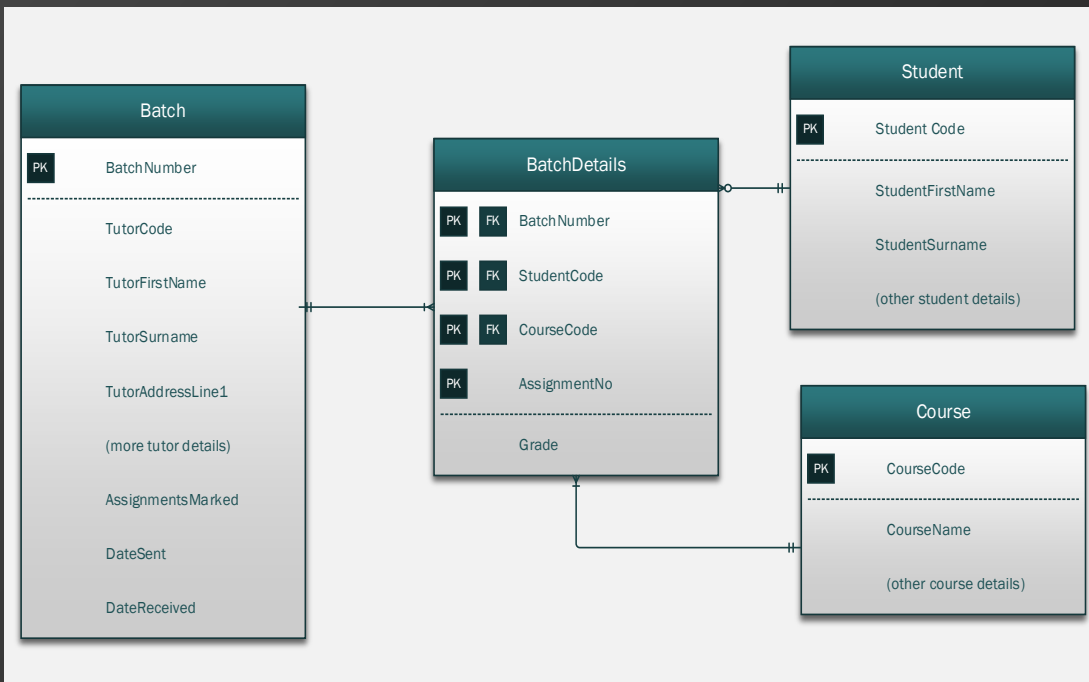
All non-key attributes must be dependent on ALL parts of the primary key



The Batch table is already in 2NF. From the BatchDetails table we take out the Student and the Course details and create two new tables.

COLLEGE BATCH MARKING SYSTEM: 3NF

No non-key attributes are dependent on other non-key attributes.



Take out the Tutor details and create a Tutor table.

DATA INTEGRITY:

The consistency and correctness of data stored in a database over its lifecycle.

- A major part of database design is ensuring that the database is protected from anything that could affect its accuracy.
- Preserving the integrity of the data in a system is an ongoing process.
- If the data is not accurate, it might as well not exist!

EXAMPLES:

- You register for an online store and mistype your surname.
- You register for an online store and type the number of your address as 23 instead of 28.
- You deposit £2000 into your bank account, but the system records the deposit as £200.

THREATS TO THE DATABASE

User Error:

Users deleting data that shouldn't be deleted or editing data incorrectly.

The data collected prior to entry into the database is inaccurate or incomplete.

An employee intentionally damages the data.

THREATS TO THE DATABASE

Software Errors:

Programming errors in the application allowing erroneous data to be introduced into the database.

Hardware Failures:

A breakdown of computer hardware. Physical damage such as flooding, loss of power, etc.

THREATS TO THE DATABASE

Malicious Damage:

The corruption of the data by authorised or unauthorised users, e.g. someone hacks into the database with the intent to damage the data, or an employee intentionally damages the data.

Concurrency Errors:

Issues brought about when one user's actions in the database interfere with another user's actions (more on this next week).

DATA INTEGRITY

Each data item & relationship must be an accurate representation of their real-world counterparts

1. **Entity Integrity:** in each table, each row is unique and has a non-null primary key.
2. **Referential Integrity:** primary/foreign key links are correct. Data in one table does not contradict the data in another table.
3. **Domain Integrity:** the data we are storing should be correct.

Consistency

Correctness

Integrity = correctness and consistency of data

CONSISTENCY VS CORRECTNESS

“The system cannot enforce truth,
only consistency” (Date, 2004)

Correct implies Consistent

Inconsistent implies Incorrect

ENTITY INTEGRITY

Consistency

Each row uniquely
represents one entity
in the database
domain
e.g. an individual student

All rows in a table
contain the same
type of information

The uniqueness is
preserved by
primary keys

Primary keys cannot
be null

ENTITY INTEGRITY

<u>Student ID</u>	First Name	Last Name	Address	*Course Code
212	Lesley	May	Rich Rd	VSCOGD123
	Peter	Day	Car St	VSCOMP122
345	Brandon	Kay	Drum Lane	VSCOFS128
122	Kai	Waylay	Guitar Avenue	VSSWEN112

- ✓ Same type of data
- ✓ Each row unique
- ✗ primary keys cannot be null

REFERENTIAL INTEGRITY

Consistency

Tables are linked by relationships

The use of primary and foreign keys to define the relationships

Foreign key value must match a primary key value in the related table

E.g. a course must exist in the course table before a student can enrol on the course

REFERENTIAL INTEGRITY

<u>Student ID</u>	First Name	Last Name	*Course Code
212	Lesley	May	VSCOGD123
333	Peter	Day	VSCOMP122
345	Brandon	Kay	VSCOFS128
122	Chloe	Way	VSSWEN112

<u>Course Code</u>	Course Description
VSCOGD123	Computer Games Dev
VSSWEN112	Software Engineering
VSCOMP128	Computer Science



ENTITY & REFERENTIAL INTEGRITY

Emp
&
Dept

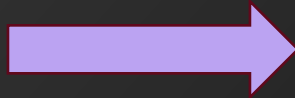
```
create table dept(  
  deptno number(2,0) PRIMARY KEY NOT NULL,  
  dname varchar2(14),  
  loc varchar2(13)  
);
```

```
create table emp(  
  empno number(4,0) PRIMARY KEY NOT NULL,  
  ename varchar2(10),  
  job varchar2(9),  
  mgr number(4,0),  
  hiredate date,  
  sal number(7,2),  
  comm number(7,2),  
  deptno number(2,0) NOT NULL REFERENCES dept(deptno)  
);
```

ENTITY
INTEGRITY

&

REFERENTIAL
INTEGRITY



Good database design



Normalisation

DOMAIN INTEGRITY

Correctness

Data capture and handling of data to prevent or at least minimise:

- Transposition errors
- Misspelling.
- Repetition of characters.

The primary defence are
Integrity Constraints:

- Validation techniques.
- Type checking.

DATA VALIDATION TECHNIQUES

Within the
field definition

- Checks each field independently
- Attribute constraints, e.g. size, type, range, presence, etc.

At table level

- Checks can be made that reference two or more fields of the table
- Transition constraints, e.g. never change married to single, it would either be divorced or widowed.

At database
level

- Transition constraints, e.g. record created in 1 table may affect another table.

TYPE CHECKING

Typing of
attributes

- Numeric
- Text
- Date
- Etc.

Type dictates
maximum size
and behaviour

- Range of acceptable values on input, e.g. a list of words or numbers between 1 and 6.
- Permittable operations on the data, e.g. you can't do arithmetic on text columns .
- How data is handled on output.

INPUT VALIDATION

- Should not accept any data that fails a validation check.
- Should use at least one validation or verification check on all data entered.

“The system cannot enforce truth,
only consistency” (Date, 2004)

VALIDATION CHECKS

- Completeness,
e.g. *jallen17@uclan.ac.uk* instead of *jallen17@uclan.ac.*
- Format, e.g. *specify a format for postcode, date, email, etc.*
- Range, e.g. *numbers between 1 and 6.*
- Presence - *NOT NULL.*
- Check Digit – e.g. *telephone numbers must start with a 0.*
- Consistency – *only specific titles are allowed, e.g. Miss, Mr, Mrs.*

SQL CONSTRAINTS

- PRIMARY KEY – ensures that each column is unique.
- FOREIGN KEY – ensures the links between the tables are enforced.
- NOT NULL – a column must contain a value.
- UNIQUE - all values in a column must be different.
- DEFAULT – sets a default value for a column if a value is not specified.
- CHECK – ensures that a value in a column satisfies a specific condition.

DATA DICTIONARIES

Title	First Name	Surname	DOB	House No	Postcode	Mobile	Email
Mr	Tony	Abbott	15/04/1995	10	PR25 2TE	07856 676767	AAbbott@uni.ac.uk
Mr	Ryan	Kegworth	22/08/1993	17	PR1 3HE	08987 678765	RKegworth@uni.ac.uk
Miss	Melanie	Kellett	01/10/1997	43	PR7 8MH	07895 564567	MKellett@uni.ac.uk

Field Name	Data Type	Field Size	Input Validation/Verification / Other Information
Title	Text	10	Acceptable values (Miss, Mr, Mrs, etc.)
First Name	Text	20	Is there anything else we could check?
Surname	Text	20	
DOB	Date/Time	Short Date	Select from calendar
House No	Text	4	Numbers only 0-9
Postcode	Text	8	Format mask to accept only valid UK postcodes
Mobile	Text	12	Numbers only 0-9
Email	Text	30	Format mask to accept only valid UK postcodes. Must be unique.

THE CHECK CONSTRAINT: AN EXAMPLE

```
create table customer(  
  custno      varchar(7) PRIMARY KEY NOT NULL,  
  title       varchar2(4),  
  firstname   varchar2(20),  
  surname     varchar2(20),  
  DoB        date,  
  houseno     varchar2(5),  
  postcode    varchar2(7),  
  mobileno    varchar2(11),  
  email       varchar2(50),  
  
  CHECK (title in ('MISS', 'MR', 'MRS'))  
);
```

CHECK CONSTRAINT EXAMPLE



INSERT INTO CUSTOMER
VALUES ('100', 'MISS', 'VALERIE', 'JONES', '1-5-2005',
'21', 'PR29YT', '0746829456', 'VALERIE2@GMAIL.COM')



INSERT INTO CUSTOMER
VALUES ('101', 'DR', 'BOB', 'SPENCER', '10-9-1978', '305',
'PR19KJ', '0734901723',
'BOBTHEDOCTOR@HOTMAIL.COM')

ORA-02290: check constraint (WKSP_JALLEN17.SYS_C00155268280) violated

THE CHECK CONSTRAINT:

ANOTHER EXAMPLE

```
create table orders(  
  orderID    varchar(7) PRIMARY KEY NOT NULL,  
  amount     number CHECK (amount > 0)  
);
```



INSERT INTO CUSTOMER VALUES ('100', 'MISS',)



INSERT INTO orders VALUES ('101', -5)

DATA TYPES IN ORACLE

CHAR (size)	consists of upper and lower case letters, numbers and special characters (+,-,%,&, etc.) size the maximum length, in characters, of the column. May not be larger than 255.
VARCHAR (size)	Variable length character string data type. Only stores the actual length of the data field, does not space fill to size specified. size must be specified. Can distinguish between NULL and an empty string.
VARCHAR2 (size)	Same as VARCHAR, except does not distinguish between NULL and an empty string.
LONG	Character data of variable length up to 65,535 characters. Only one long column per table.
NUMBER	Number values consisting of digits 0 -- 9, values may be 38 digits wide.
NUMBER (w,d)	Number values with decimal places specified
DECIMAL (w,d)	Stores numbers with up to 22 decimal digits. w - specifies the total number of digits. d - specifies the number of decimal places. (i.e. decimal (4,2) will allow a max. number of 99.99 to be inserted.)
INTEGER	Stores numbers with 10 or fewer digits, digits to the right of the decimal point are truncated.
SMALLINT	Stores numbers with 5 or fewer digits.
DATE	Date values - usual form dd-mmm-yy i.e. 30-OCT-98

SUMMARY

- Database Integrity is concerned with the consistency and correctness of the data in a database throughout its lifecycle.
- A combination of Entity, Referential, and Domain (data) Integrity is used to preserve the data.
- Data validation techniques are used to ensure the data entering the database is correct.

RESOURCES

- Richie, C. (2008). *Database Principles and Design*. 3rd Ed. Thompson Learning
- https://www.w3schools.com/sql/sql_check.asp