

# INFO90002 – example Data Dictionary - the TV database

## List of entities

Entities	Entity Type	Primary Key	# Attributes
CHANNEL	independent	channel_id	3
PROGRAM	independent	program_id	11
GENRE	dependent	program_id, genre_name	2
STAR	independent	star_id	2
CAST	dependent	star_id, program_id	3
CONSUMER_ADVICE	independent	code	2
WARNING	dependent	program_id, code	2
BROADCAST	independent	SMS_code	9
CUSTOMER	independent	phone_no	1
REMINDER	dependent	phone_no, SMS_code	3

## Entity 'CHANNEL'

### Description

This entity represents the TV station which is broadcasting programs on the channel in question.

### Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PK	channel_id	SmallInt	YES	YES	While this primary key is a unique integer, it is also the number by which the TV station is best known on a TV tuner. E.g. '9' for Channel Nine, '2' for The ABC, etc.
	channel_name	Varchar(30)	YES	NO	The name of the TV station itself e.g. 'Channel NINE' or 'ABC 1' or 'SBS'.
	analog-digital	Char(1)	YES	NO	A single character variable with the letter 'A' if the channel broadcasts in Analog, or 'D' if the channel broadcasts in Digital format.

## Entity 'PROGRAM'

### Description

For each new program (movie, news, episode of a soap, etc) entered into the database. There will be one record for each episode of an ongoing program (e.g. a soap opera or the nightly news), but not for 'repeat' broadcasts'. The entity PROGRAM holds 'static' data about a program: those attributes which do not vary from showing to showing.

### Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PK	program_id	Integer	YES	YES	This is the primary-key of PROGRAM, generated by the DBMS, unique in the lifetime of this database. If a movie is shown three times during the year, there will only be one record in this entity, the details of which will be used by all three broadcasts.
	title	Varchar(70)	NO	NO	Holds the actual title of the program e.g. the name of a movie, or 'ABC News' a regularly timetabled news program.
	description	Varchar(255)	NO	NO	A description of the program. Currently it may be up to 255 characters. This might be too small - should clarify with the client to see what they have in mind for this field. It's too small to hold a 'review', but it is large enough to hold the small descriptions one generally sees in published TV guides.
	country	Varchar(50)	NO	NO	The name of the country where the program was produced. Some programs are collaborations between companies or organisations across two or more countries. In such cases, this field will hold just the country of the primary collaborator. This field may need to become a multi-valued field (hence another database entity) in the future, if collaborations are common or become more common. Yet another issue to be taken up with the client.
	year_made	Year(4)	NO	NO	Just the four-digit year that the program was produced in. e.g. '2006', for a movie released in 2006.
	classification	Char(5)	NO	NO	The formal classification assigned to the Program by some authority, for the country concerned, depicting community standards regarding what children, teenagers and adults can expect to see within any given program. So, in this

Key	Attribute	Data Type	Not Null	Unique	Description
					case, the Australian classification system, defined, and enforced by the Australian Federal Government. This attribute may have one of the following values: 'G', 'PG', 'M', 'MA15+', 'AV15+'. There is potential for this set of values to change in the near future, as authorities move to a unified classification system which covers, TV, Movies and Computer Video Games.
	high_definition	Char(1)	NO	NO	Television programs at this point in time, can be filmed in a number of different formats regarding the quality of the image (scan-lines, and pixels per line). High Definition is a newer format, showing more scan-lines and more pixels per line, than earlier traditional resolutions. This field holds either a 'T' for True, or 'F' for False- regarding whether or not the program was filmed in High-Definition.
	wide_screen	Char(1)	NO	NO	Like High-definition, wide-screen is also to do with the visual properties of the filming when a program was made. The pixel ratio for wide-screen is 9 (wide) by 6 (high), regularly seen in cinemas. The more tradition ratio, as with conventional computer screens is 4 (wide) by 3 (high). The media industry seems to be currently in transition between these two formats, as more and more TV sets and computer screens are coming out in the wide- screen format (9x6), If a wide-screen program is shown on a 4x3 ratio screen, it generally has a band of black along the top and bottom the screen.
	sub_titled	Char(1)	NO	NO	May hold either the value 'T' for True, or 'F' for False- indicating whether or not the program comes with sub-titles (in English - in the case of Australian TV)
	episode_no	Smallint	NO	NO	In the case of a soap Opera or a similar genre that has multiple episodes, each episode will have its own record in this PROGRAM entity, and therefore, this attributes tells us the number of the episode this record actually represents. E.g. Different episodes may star different people, so it is sensible to store records here at the 'episode level'. If the program doesn't have multiple episodes, it should be left as NULL, rather than '1'- so that the 'total_episodes' need not be set either.
	total_episodes	Smallint	NO	NO	In the case of a soap Opera or a similar genre that has multiple episodes, each episode will have its own record in this PROGRAM entity, and therefore this attribute tells us the total number of episodes that were ever made for the series. We can therefore calculate from the value in this attribute together with 'episode_no', whether or not the episode being shown is the 'Final' episode - one of the requirements from the Client.

## Entity 'GENRE'

### Description

Unlike the STAR entity, this entity has been created by simply removing the 'multi-valued' attribute 'genre' from the ER diagram, calling it 'genre\_name', and using its values themselves as a part of the concatenated primary-key of this entity GENRE. While it could have been done similarly to STAR, by generating a new integer primary-key called something like 'genre\_id', and using that in an associative entity between PROGRAM and GENRE, the short names that appear as values in 'genre\_name' (e.g. 'Comedy'), and the lack of other attributes related to genre of any consequence, led me to do it this way.

### Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PFK	program_id	Integer	YES	NO	A foreign-key that makes up one part of the two-part concatenated primary-key, it comes from the entity PROGRAM, about which the value in 'genre_name', describes.
PK	genre_name	Varchar(20)	YES	NO	The second-part of the concatenated primary-key of this entity. Up to 20 characters in length, the values of this attribute include, but are not limited to: Romance, Sitcom, Drama, Comedy, Entertainment, Reality, Cooking, GameShow, Travel, Animation, Religion, SoapOpera, Documentary, News, KidsShow, Sport, Education and Lifestyle.

## Entity 'STAR'

### Description

The entity holds the name of Stars and Presenters of programs. Together with CAST it offers flexibility in holding members of a particular program, such as the Director of a movie for example. It currently only

holds the 'name' of such people, but offers the chance to expand it with a number of other possible attributes, should the clients need arise.

### Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PK	star_id	Integer	YES	YES	The name of a Star or a Presenter (or any other Role) of a person performing in a program held in the entity PROGRAM
	Name	Varchar(50)	YES	NO	A single attribute to hold the Stars name, as people in the industry are usually known by the full name. E.g. 'Nicole Kidman'; 'Madonna'.

## Entity 'CAST'

### Description

This is an Associative Entity between STAR and PROGRAM, allowing for multiple stars per program, using their respective primary keys, without duplicating any other details of STAR (i.e. their 'name'), nor of PROGRAM.

### Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PFK	star_id	Integer	YES	NO	A foreign_key, coming from the entity STAR.
PFK	program id	Integer	YES	NO	A foreign_key, coming from the entity PROGRAM.
	role	Varchar(50)	YES	NO	This attribute holds the 'role' of this STAR in this PROGRAM. Example values of role are 'Presenter' and 'Star', but role could be used to hold many other useful values.

## Entity 'CONSUMER\_ADVICE'

### Description

Unlike a Program's formal 'classification', of which there can only be one, a program can carry several 'Consumer Advice' warnings, such as the example Program given in the client's specification, which has three. Therefore, CONSUMER\_ADVICE has been declared as a separate entity in this data model, with a related associate entity (called WARNING) between it and PROGRAM, to give the necessary ability to represent 'any number' of warnings, for a given single Program.

### Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PK	Code	Char(1)	YES	YES	This is a single character field, which holds a single letter to represent the value in its fellow attribute 'description'. i.e. 'A' stands for 'Adult themes'; 'L' stands for 'frequent coarse Language'; 'N' stands for 'Nudity'; 'S' stands for 'Sexual references'.
	description	Varchar(50)	NO	NO	Each record in this entity holds the text of the specific 'Consumer Advice Warnings'. Currently, the possible values held here in 'description' are: 'Adult themes'; 'Frequent coarse language'; 'Sexual references'; 'Nudity' - but it could clearly hold many others, should the need arise in the future.

## Entity 'WARNING'

### Description

This is an Associate Entity between PROGRAM and CONSUMER\_ADVICE, allowing for multiple consumer advice warnings per program, using the respective primary keys of those two entities, but without duplicating any other attributes of either PROGRAM or CONSUMER\_ADVICE. It currently has no other attributes beyond the foreign-keys.

### Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
-----	-----------	-----------	----------	--------	-------------

PFK	program_id	Integer	YES	NO	A foreign-key, coming from the entity PROGRAM.
PFK	Code	Char(1)	YES	NO	A foreign-key, coming from the entity CONSUMER_ADVICE, it is single-character in size, having values of either: 'A', 'L', 'N' or 'S'- as described in that entities part of this dictionary.

## Entity 'BROADCAST'

### Description

Represents the event in which a Program is sent over the airwaves. Holds attributes that are specific to that one event.

### Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PK	SMS_code	Bigint	YES	YES	The SMS_Code itself is required to be unique, and seems like a good choice for the primary-key of this entity. i.e. An individual SMS_code is needed for each broadcast program, so that appropriate reminder messages can be sent out to participating customer/TV-viewers. However, using it as primary- key does govern that its value shall-not-be reused during the lifetime of the database, which is beyond the 7-days of the TV Guide, as the database records themselves are used to calculate the 'repeat' attribute, held further down in this entity. On the other hand, letting the database generate the primary-key, does guarantees us that no two SMS- Codes will ever be the same, with no work on any humans part in the process.
FK	channel_id	Smallint	YES	NO	A foreign-key coming from the entity named 'CHANNEL', and representing the TV station that is broadcasting the specific program at this time.
FK	program_id	Integer	YES	NO	A foreign-key coming from the entity named 'PROGRAM', and representing the program that is being broadcast at this time.
	start_day	Char(1)	NO	NO	The day on which the broadcast of the program starts. Note: Some programs broadcasts will begin in one day_, but finish within another.
	start_date	Date	NO	NO	The full date (YYYY:MM:DD) on which the broadcast of the program starts. Note: Some programs broadcasts will begin on one date, but finish within another.
	start_time	Time	NO	NO	The time (hh:mm:ss) on which the broadcast of the program starts.
	actually_happened	Char(1)	NO	NO	This is an attribute whose value can only be input to the database, after the scheduled broadcast has passed in time. It holds a value of 'T' for 'True' if the scheduled program did take place or 'F' for False, if it failed to materialise. This field may be useful to the client in answering customer complaints, in the event of TV stations not broadcasting programs they said they would.
	duration	Smallint	NO	NO	The actual time period (in minutes) for which the broadcast program is expected to run. This time comes from the TV station programmers themselves, and includes the time for advertisements, station promos and community announcements, that are likely to be broadcast during the time period that the program will be televised.
	repeat	Char(1)	NO	NO	The value of the attribute may be either 'T' for True or 'F' for False, which can be determined programmatically (a derived field - as per the ER diagram of the model), to show potential TV viewers, whether or not the program has been broadcast before by the Channel. ie. All broadcast records need to be kept for the life of the database, to enable this attribute to be correctly determined.

## Entity 'CUSTOMER'

### Description

The only attribute that the system currently needs to store for the entity CUSTOMER, is the customers phone number. However, by making CUSTOMER an entity is prudent, as it allows for several sensible expansions of the system, sometime in the near future (see 'Potential Issues').

### Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PK	phone_no	Varchar(18)	YES	YES	This holds the customer phone number- the one that the client called from - which is assumed to be the one upon which they want the returned SMS reminder/s. Note: by having CUSTOMER as an entity, I have allowed for easy expansion of the number of attributes that could be held for a customer, including a second (or more) other phone numbers upon which they might

					prefer the reminding SMS message to be sent (see 'Potential Issues').
--	--	--	--	--	---

## Entity 'REMINDER'

### Description

This entity has a record for all of the SMS reminders that have to be output by the system. It is an Associate- Entity between BROADCAST and CUSTOMER, therefore allowing for the one customer to get multiple SMS messages for different programs, and for the same message about the same broadcast to be sent to multiple customers.

### Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PFK	phone_no	Varchar(18)	YES	NO	A foreign-key, coming from the entity CUSTOMER
PFK	SMS_code	BigInt			A foreign-key, coming from the entity BROADCAST.
	lead_time	Smallint	NO	NO	Currently, via the client specification, the amount of time before a broadcast that a customer is to receive their requested reminder, is not specified. Via client consultation, it might be set at a fixed figure- say, 5 minutes before the broadcast- or it could be made a system feature to allow the customer themselves to specify such a 'lead-time' in minutes, for when they want to be reminded. This attribute holds the 'lead- time' in minutes, in either case.