

Normalización – Proyecto 1

habitaciones	
P *	numero INTEGER
F *	tipo VARCHAR2 (50 CHAR)
habitaciones_PK (numero)	
habs_tipo_habitacion_FK (tipo)	

Habitaciones persiste el numero asignado a una habitación y una foreign key que referencia al tipo de habitación que es. Sus llaves primas son el número y su no primas el tipo. Por tanto, su nivel de normalización es BCFN.

tipos_habitacion	
P *	nombre VARCHAR2 (50 CHAR)
*	precio_base NUMBER (12,2)
*	capacidad INTEGER
tipos_habitacion_PK (nombre)	

Tipos habitación persiste el nombre de un tipo, su precio base y la capacidad de hospedaje que tiene. Sus llaves primas son el nombre y sus no primas el precio base y la capacidad. Por tanto, su nivel de normalización es BCFN.

dotaciones	
P *	nombre VARCHAR2 (50 CHAR)
*	valor_agregado NUMBER (12,2)
dotaciones_PK (nombre)	

Dotaciones es cada uno de ítems presentes en una habitación, compuestos por un nombre y un valor que por su presencia es agregado al valor de hospedaje por noche de dicha habitación. Sus llaves primas son el nombre y sus no primas el valor agregado. Por tanto, su nivel de normalización es BCFN.

dotacion_habitacion	
PF *	tipo_habitacion VARCHAR2 (50 CHAR)
PF *	dotacion VARCHAR2 (50 CHAR)
dotacion_hab_PK (tipo_habitacion, dotacion)	
dothab_tipo_habitacion_FK (tipo_habitacion)	
dothab_dotacion_FK (dotacion)	

Dotación habitación es la relación entre que dotación esta un determinado tipo de dotación. Persiste el tipo de habitación y el nombre de la dotación presente. Sus llaves candidatas son el tipo de habitación y dotación que a su vez son sus llaves primas y esta tabla no tiene llaves no primas. Por tanto, su nivel de normalización es BCFN.

reservas_estancias	
P *	id INTEGER
*	inicio TIMESTAMP
*	fin TIMESTAMP
*	cantidad_huespedes INTEGER
F *	cliente INTEGER
reservas_estancias_PK (id)	
reservas_estancias_usuarios_FK (cliente)	

Reservas de estancias persiste un identificador único, una fecha de inicio, una fecha de fin, una cantidad de huéspedes y una foreign key que referencia al cliente. Su PK es el id, que a su vez es su única llave prima, por otros lados sus llaves no primas son las fechas de inicio, fecha de fin, cantidad de huéspedes y el cliente. Dado que ninguna de llaves mencionadas depende del atributo cliente, entonces esta table cuenta con nivel de normalización BCFN.

reserva_habs	
PF *	reserva_id INTEGER
PF *	habitacion_numero INTEGER
reserva_habs_PK (reserva_id, habitacion_numero)	
rsv_reserva_FK (reserva_id)	
rsv_habitacion_FK (habitacion_numero)	

Reserva habs es una relación entre que habitaciones fueron asignadas a determinada reserva. Es por ello que cuenta con la llave candidata compuesta del id de la reserva y el numero de la habitación que a su vez son sus únicos campos y son primos. Por tanto, el nivel de normalización es BCFN.

estancias	
PF *	id INTEGER
estancias_PK (id)	

Una estancia es la confirmación de una reserva se efectuó. Es decir, los huéspedes se alojaron en el hotel. Por lo que solo tiene un campo

que es primo y a su vez es su PK y se trata de un id, que correspondería al mismo de la reserva. Por tanto, sería PK y FK al mismo tiempo y su nivel de normalización es BCFN.

ocupaciones		
PF *	habitacion_numero	INTEGER
PF *	estancia_id	INTEGER
P *	fecha	TIMESTAMP
ocupaciones_PK (habitacion_numero, estancia_id, fecha)		
ocp_habitacion_FK (habitacion_numero)		
ocp_estancia_FK (estancia_id)		

Ocupaciones es la relación entre la estancia y las habitaciones reales en que se encuentran los huéspedes alojados por cada día de presencia. Es por ello por lo que su llave candidata se compone de los campos primos número de habitación, el id de la estancia y la fecha del día que se ocupó. Dado que cada campo es primo y no hay campos primos el nivel de normalización es BCFN.

roles		
P *	nombre	VARCHAR2 (50 CHAR)
roles_PK (nombre)		

Roles persiste el nombre de un rol al que puede aspirar un usuario. Esta tabla solo contiene un campo primo que es su misma PK y se trata de nombre. Por tanto, su nivel de normalización es BCFN.

usuarios		
*	nombre	VARCHAR2 (100 CHAR)
*	apellido	VARCHAR2 (100 CHAR)
*	edad	INTEGER
P *	documento	INTEGER
*	email	VARCHAR2 (100 CHAR)
*	tipo_documento	VARCHAR2 (50 CHAR)
F *	rol	VARCHAR2 (50 CHAR)
usuarios_PK (documento)		

Usuarios son todos aquellas que interactúan dentro o con el hotel. Esta tabla se compone de los campos primos documento y los no primos nombre, apellido, edad, email, tipo documento y una foreign key que apunta al rol de dicho usuario. Ya que ningún de los campos no primos depende de otro primo y tampoco hay presencia de foreign keys compuestas el nivel de normalización es BCFN.

planes_de_consumo		
P *	nombre	VARCHAR2 (100 CHAR)
	descuento	NUMBER (5,2)
	mínimas_noches	INTEGER
	valor	NUMBER (12,2)
F	vigencia_id	INTEGER
F	restriccion_id	INTEGER
planes_de_consumo_PK (nombre)		
plc_restriccion_dias_FK (restriccion_id)		
plc_consumo_vigencias_FK (vigencia_id)		
planes_de_consumo_IDX (restriccion_id)		
planes_de_consumo_IDXv1 (vigencia_id)		

Planes de Consumo persiste el nombre, el descuento, las mínimas noches, el valor y tiene 2 foreign keys los cuales son el ID de la vigencia y el ID de las restricciones según los días. Sus llaves primas son el nombre y las no primas son el descuento, las mínimas noches, el valor, el ID de la vigencia y el ID de las restricciones según los días. Por lo tanto, su nivel de normalización es BCFN.

vigencias		
P *	id	INTEGER
*	inicio	TIMESTAMP
*	final	TIMESTAMP

Vigencia son las fechas en las cuales se puede usar un plan de consumo. De una vigencia persiste un identificador único, una fecha de inicio y una fecha de fin. Su PK es el id, que a su vez es su única llave prima. Por otro lado, sus llaves no primas son las fechas de inicio y de fin. Debido a que ninguna de las llaves anteriormente mencionadas depende de algún atributo no primo, entonces esta tabla está en el nivel de normalización de BCFN.

restriccion_dias		
P *	id	INTEGER
	lunes	INTEGER
	martes	INTEGER
	miercoles	INTEGER
	jueves	INTEGER
	viernes	INTEGER
	sabado	INTEGER
	domingo	INTEGER
restriccion_dias_PK (id)		

Restricción días son los días sobre los cuales un plan de consumo tiene vigencia. De restricción días persiste un identificador único, y los días de la semana de manera individual. Su PK es el ID, que a su vez es su única llave primaria. Por otro lado, sus llaves no primas son los días de la semana de manera individual. Debido a que ninguna de las llaves anteriormente

mencionadas depende de algún atributo, esta tabla está en el nivel de normalización de BCFN.

consumos	
P * id	INTEGER
* valor_total	NUMBER (12,2)
* fecha	TIMESTAMP
F * servicio_id	INTEGER
F * estancia_id	INTEGER
consumos_PK (id)	

Consumos es el registro de lo que un cliente del hotel gasto en determinado servicio. Se compone de una llave prima y PK de la tabla que es el id, las llaves no primas del valor total, fecha y las FK al servicio y la estancia registrada para dicho usuario. Ya que ningún atributo no primo depende de otro no primo y no hay llaves foráneas compuesta esta tabla se encuentra con nivel de normalización BCFN.

items	
P * id	INTEGER
* nombre	VARCHAR2 (100 CHAR)
* valor	NUMBER (12,2)
Items_PK (id)	

Ítems es cada uno de los elementos vendibles o consumibles dentro del hotel. Se compone de la llave prima y PK id y de las llaves no primas nombre y valor, dado que cada atributo no primo es independiente esta tabla se encuentra en BCFN.

items_consumidos	
PF * consumo_id	INTEGER
PF * item_id	INTEGER
items_consumidos_PK (consumo_id, item_id)	
consumo_FK (consumo_id)	
consumo_items_FK (item_id)	

Ítems consumidos es la relación de que ítems fueron adquiridos por un usuario en un consumo. Se compone de los campos id del consumo e id del ítem que a su vez en conjunto forman la llave candidata y son primos. Dado que no hay atributos no primos el nivel de relación de esta tabla es BCFN.

productos	
* origen	VARCHAR2 (20)
PF * item_id	INTEGER
productos_PK (item_id)	
prod_item_FK (item_id)	

Productos es un ítem adquirible y/o consumible para los usuarios. Se compone de los campos primos id del item que es su PK y es FK de la tabla ítems, sus campos no primos son el origen de dicho producto. En este todos los campos son atómicos, no hay dependencias parciales ni transitivas y las llaves foráneas no son compuestas por tanto el nivel de normalización es BCFN.

servicios_incluidos	
PF * plan_de_consumo	VARCHAR2 (100 CHAR)
PF * servicios_id	INTEGER
* descuento	NUMBER (5,2)
servicios_incluidos_PK (plan_de_consumo, servicios_id)	
ser_inc_plc_FK (plan_de_consumo)	
ser_incl_servicios_FK (servicios_id)	

Servicios Incluidos es una relación entre la clase servicios y la clase de plan de consumo y son todos los servicios que tiene un plan de consumo. Por esto, la llave candidata se compone del ID de servicios y el nombre del plan de consumo la cual termina siendo su PK. Además, se tiene una llave no prima que es el descuento. Dado a que ninguna de los atributos tiene dependencia parciales o transitivas y no hay llaves claves candidatas y compuestas que se intersectan, se puede decir que la tabla está en BCFN.

productos_incluidos	
PF * plan_de_consumo	VARCHAR2 (100 CHAR)
PF * producto_id	INTEGER
* cantidad	NUMBER (12,2)
productos_incluidos_PK (plan_de_consumo, producto_id)	
prod_incl_plc_FK (plan_de_consumo)	
prod_incl_producto_FK (producto_id)	

Productos Incluidos es una relación entre la clase Productos y la clase de plan de consumo y son todos los productos que tiene un plan de consumo. Por esto, la llave candidata se compone del ID de los productos y el nombre del plan de consumo la cual termina siendo su PK. Además, se tiene una llave no prima que es la cantidad. En vista de que ninguna de los atributos tiene dependencia parciales o

transitivas y no hay llaves claves candidatas y compuestas que se intersectan, se puede decir que la tabla está en BCFN.

servicios	
P *	id INTEGER
*	nombre VARCHAR2 (50 CHAR)
servicios_PK (id)	

Servicios es el registro de todos los servicios con los cuales cuenta el hotel. De servicios Persiste un identificar único que es su PK y es su llave primaria. También se tiene nombre, el cual es el nombre del servicio, y es una llave no

prima. Debido a que ninguna de los atributos tiene dependencia parciales o transitivas y no hay llaves claves candidatas y compuestas que se intersectan, se puede decir que la tabla está en BCFN.

spa	
*	capacidad INTEGER
*	apertura TIMESTAMP
*	cierre TIMESTAMP
*	franja INTEGER
PF *	servicio_id INTEGER
spa PK (servicio id)	

Spa persiste los datos de cada uno de los spas disponibles en el hotel. Específicamente los campos primos del id de servicio que es un FK de la tabla servicios y PK del spa. Los campos no primos relacionados a la capacidad, hora de apertura, hora de cierre, franja

que la unidad de tiempo reservable. En esta tabla se evidencia campos atómicos y cero dependencias parciales o transitivas, así como claves foráneas compuestas por tanto su nivel de normalización es BCFN.

servicios_spa	
PF *	spa_id INTEGER
PF *	Items_id INTEGER
*	duracion INTEGER
servicios_spa_PK (spa_id, Items	
spa_FK (spa_id)	
spa_Items FK (Items_id)	

Servicios spa son todos aquellos servicios brindados por en un spa. Se compone de los campos primos id del spa y id del ítem que a su vez en conjunto son la llave candidata de la tabla y claves foráneas de las tablas de servicios e ítems, también se compone de un campo no primo que es la duración de dicho servicio. En este caso cada campo es atómico, no hay dependencias de ningún tipo y tampoco claves foráneas compuestas.

bares	
*	capacidad INTEGER
*	estilo VARCHAR2 (50)
PF *	servicio_id INTEGER

Bares persiste la información de cada uno de los bares del hotel. De estos nos interesa la información organizada por el campo primo id del servicio que es la PK de la tabla y la FK de la tabla servicios, así como de los campos no primos de capacidad y estilo. Cada uno de los atributos es atómico, no hay relaciones parciales o transitivas y tampoco claves foráneas compuestas. Es por esto que el nivel de normalización es BFCN.

bares_productos	
PF *	bar_id INTEGER
PF *	producto_id INTEGER
bares_productos_PK (bar_id, p	

Bares productos son todos aquellos productos ofrecidos por un bar. La tabla se compone de los campos primos id del bar e id de producto que en conjunto forman la llave candidata de la tabla. Dado que no hay

atributos no primos y que cada atributo primo es independiente esta tabla se encuentra en BCFN.

restaurantes	
*	capacidad INTEGER
*	estilo VARCHAR2 (50)
PF *	servicio_id INTEGER

Restaurantes persiste la información de cada uno de los restaurantes del hotel. De estos nos interesa la información compuesta por el campo primo id del servicio que es la PK de la tabla y la FK de la tabla

servicios, así como de los campos no primos de capacidad y estilo. Cada uno de los atributos es atómico, no hay relaciones parciales o transitivas y tampoco claves foráneas compuestas. Por tanto, el nivel de normalización es BCFN.

restaurante_prods	
PF *	restaurante_id INTEGER
PF *	producto_id INTEGER
restaurante_prods_PK (restaurante_id, producto_id)	

Restaurante prods son todos aquellos productos ofrecidos por un restaurante del hotel. La tabla se compone de los campos primos id del restaurante e id de producto que en conjunto forman la llave candidata de la tabla. Dado que no hay atributos no primos y que cada atributo primo es independiente esta tabla se encuentra en BCFN.

tiendas	
*	tipo VARCHAR2 (50)
PF *	servicio_id INTEGER

Tiendas persiste la información de cada una de las tiendas del hotel. De estas nos interesa la información compuesta por el campo primo id del servicio que es la PK de la tabla y la FK de la tabla servicios, así como del campo no primos tipo de tienda. Cada uno de los atributos es atómico, no hay relaciones parciales o transitivas y tampoco claves foráneas compuestas. Por tanto, el nivel de normalización es BCFN.

tiendas_productos	
PF *	tienda_id INTEGER
PF *	producto_id INTEGER
tiendas_productos_PK (tienda_id, producto_id)	
tiendas_prod_tienda_FK (tienda_id, producto_id)	
tiendas_prod_producto_FK (tienda_id, producto_id)	

Tiendas productos son todos aquellos productos ofrecidos por una tienda del hotel. La tabla se compone de los campos primos id de la tienda e id de producto que en conjunto forman la llave candidata de la tabla. Dado que no hay atributos no primos y que cada atributo primo es independiente esta tabla se encuentra en BCFN.

reservas_servicios	
P *	fecha TIMES
*	duracion INTEGER
PF *	numero_habitacion INTEGER
PF *	servicio_id INTEGER
reservas_servicios_PK (numero_habitacion, fecha)	
resv_sev_habitacion_FK (numero_habitacion)	
resv_sev_servicio_FK (servicio_id)	

Reservas servicios son aquellas reservaciones para poder disfrutar de un servicio del hotel que realizan los huéspedes. Esta tabla se compone de los campos primos id del servicio, numero de la habitación y fecha de la reservación, así como también del campo no primo duración. Este caso cada atributo es atómico, no se observan dependencias de ningún tipo y las claves foráneas no son compuestas. Por tanto, el nivel de normalización es BCFN.

reserv_serv_items	
PF *	Item_id INTEGER
PF *	numero_habitacion INTEGER
PF *	servicio_id INTEGER
PF *	reserva_fecha TIMESTAMP
reserv_serv_items_PK (numero_habitacion, servicio_id, reserva_fecha)	
reserv_serv_items_items_FK (Item_id)	

Reserv serv items es la relación entre que ítems hacen parte de la reservación del servicio, creada para el caso de salones que en su reservación tiene equipamiento reservado u otros casos excepcionales. En este caso la tabla se compone de los campos primos id del ítem como FK de la tabla ítems y numero de habitación, id del servicio y fecha de la reserva como FK de la tabla reservas servicios. En conjunto todos los campos forman la llave candidata. En este cada campo es atómico, no hay dependencias ni parciales ni transitivas, pero una de las claves foráneas si es compuesta por lo que nivel de normalización es 3N.

salones	
* capacidad	INTEGER
* costo_hora	NUMBER (12,2)
* tiempo_limpieza	INTEGER
PF * servicio_id	INTEGER
* tipo_salones	VARCHAR2 (100 CHAR)
salones_PK (servicio_id)	

Salones son los salones de reuniones. De esta tabla, la llave candidata es el ID que hereda de la clase servicio la cual es una PK, una FK y una llave primaria, y las llaves no primas son la cantidad, el costo por hora, el tiempo de limpieza y el tipo de salón. Debido a que ninguna de los atributos tiene dependencia parciales o transitivas y no hay llaves claves candidatas y compuestas que se intersectan, se puede decir que la tabla está en BCFN.

prestamos	
* utensilio	VARCHAR2 (50 CHAR)
devuelto	VARCHAR2 (5 CHAR)
* condicion	VARCHAR2 (30 CHAR)
F * servicio_id	INTEGER
P * id	INTEGER

Prestamos es la tabla la cual dice cuales utensilios se pueden prestar según el hotel. De esta tabla su PK es el ID, que a su vez es la llave primaria. También se tiene que las llaves no primas son el utensilio, la condición de entrega, el estado de devuelto, y un FK que se hereda de la clase servicio. Debido a que ninguna de los atributos tiene dependencia parciales o transitivas y no hay llaves claves candidatas y compuestas que se intersectan, se puede decir que la tabla está en BCFN.

indumentarias	
* elementos	INTEGER
* valor_base	NUMBER (12,2)
* valor_elemento	NUMBER (12,2)
F * servicio_id	INTEGER
P * id	INTEGER
indumentarias_PK (id)	

Indumentaria hace referencia al servicio de Lavado/Planchado/Embolado. De esta clase persiste su PK que es un Identificar único que su vez es la llave primaria y las llaves no primas son los elementos, el valor base, el valor del elemento y la FK que es el ID que se hereda de la clase servicios. Debido a que ninguna de los atributos tiene dependencia parciales o transitivas y no hay llaves claves candidatas y compuestas que se intersectan, se puede decir que la tabla está en BCFN.

piscinas	
* capacidad	INTEGER
* apertura	TIMESTAMP
* cierre	TIMESTAMP
* valor	NUMBER (12,2)
* profundidad	NUMBER (12,2)
PF * servicio_id	INTEGER
piscinas_PK (servicio_id)	

Piscinas hace referencia al tipo de piscina con el cual cuenta el hotel. De esta tabla, la llave cantidad es el ID que hereda de la clase servicio la cual es una PK, una FK y una llave primaria, y las llaves no primas son la capacidad, las horas de apertura y cierre, el valor y la profundidad. Debido a que ninguna de los atributos tiene dependencia parciales o transitivas y no hay llaves claves candidatas y compuestas que se intersectan, se puede decir que la tabla está en BCFN.

internet	
* capacidad	NUMBER (6,2)
* valor_por_dia	NUMBER (12,2)
* inicia	TIMESTAMP
* finaliza	TIMESTAMP
PF * servicio_id	INTEGER
internet_PK (servicio_id)	

Internet hace referencia al internet con el que cuenta el hotel. De esta tabla, la llave cantidad es el ID que hereda de la clase servicio la cual es una PK, una FK y una llave primaria, y las llaves no primas son la capacidad, el valor por día, el tiempo de inicio y finalización. Debido a que ninguna de los atributos tiene dependencia parciales o transitivas y no hay llaves claves candidatas y compuestas que se intersectan, se puede decir que la tabla está en BCFN.

gimnasios	
* capacidad	INTEGER
* apertura	TIMESTAMP
* cierre	TIMESTAMP
* valor	NUMBER (12,2)
PF * servicio_id	INTEGER
gimnasios_PK (servicio_id)	

Gimnasio hace referencia al gimnasio con el que cuenta el hotel. De esta tabla, la llave cantidad es el ID que hereda de la clase servicio la cual es una PK, una FK y una llave primaria, y las llaves no primas son la capacidad, el valor, la hora de apertura y de cierre. Debido a que

ninguna de los atributos tiene dependencia parciales o transitivas y no hay llaves claves candidatas y compuestas que se intersectan, se puede decir que la tabla está en BCFN.

gym_maqs	
PF *	maquina_nombre VARCHAR2 (50 CHAR)
PF *	gimnasio_id INTEGER
gym_maqs_PK (maquina_nombre, gimnasio_id)	
gym_maqs_maquina_FK (maquina_nombre)	
gym_maqs_gimnasio_FK (gimnasio_id)	

Gym maqs es una relación entre la clase gimnasios y la clase máquinas y son todas las maquinas que tiene el gimnasio.

Por esto, la llave candidata se compone del ID de gimnasio y el nombre de la máquina, la cual termina siendo su PK.

Además, no se tiene llaves no primas. Dado a que ninguna de los atributos tiene dependencia parciales o transitivas y no hay llaves claves candidatas y compuestas que se intersectan, se puede decir que la tabla está en BCFN.

maquinas	
P *	nombre VARCHAR2 (50 CHAR)
maquinas_PK (nombre)	

Maquinas es cada una de las maquinas que un gimnasio podría albergar. Se compone de un solo campo que es primo y PK de la

tabla, nombre. Ya que este campo es atómico no hay ninguna dependencia y tampoco claves foráneas el nivel de normalización es BCFN.

CONCLUSION:

De una forma general se podría decir que la base de datos se encuentra en BCFN, sin embargo al tener un tabla con nivel de normalización 3, este es el nivel real de la BD.