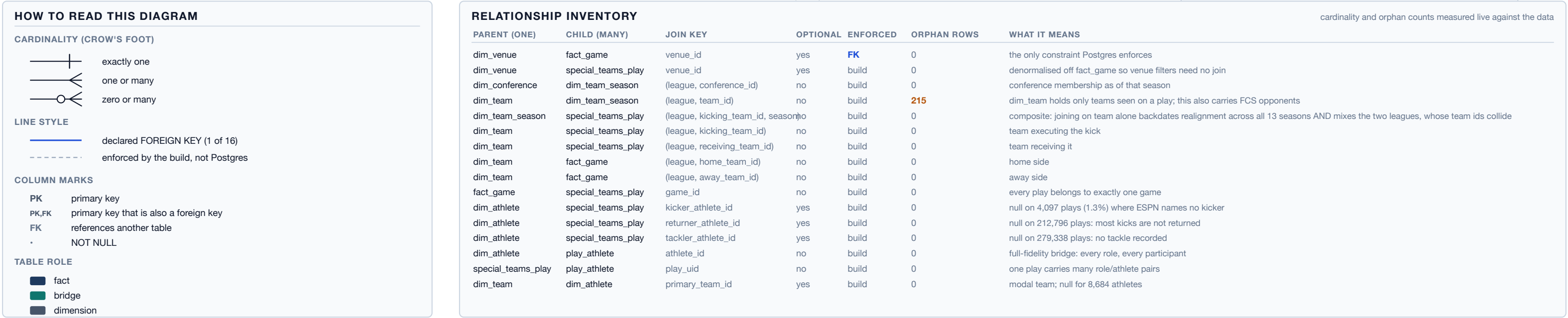


Entity Relationship Diagram — Crow's Foot notation · PostgreSQL database pbp · schema pbp · kickoffs, punts, field goals and the conversion family



Every relationship above is 1:N. The only constraint Postgres actually enforces is `fact_game.venue_id → dim_venue`; the rest are invariants of the build scripts, which is why they are dashed. Sheet 2 carries the scrimmage fact, which hangs off the same four dimensions.

Play-by-Play Warehouse — Scrimmage

Entity Relationship Diagram — Crow's Foot notation · PostgreSQL database pbp · schema pbp · rushes, passes, sacks, penalties — and the drives that contain both facts

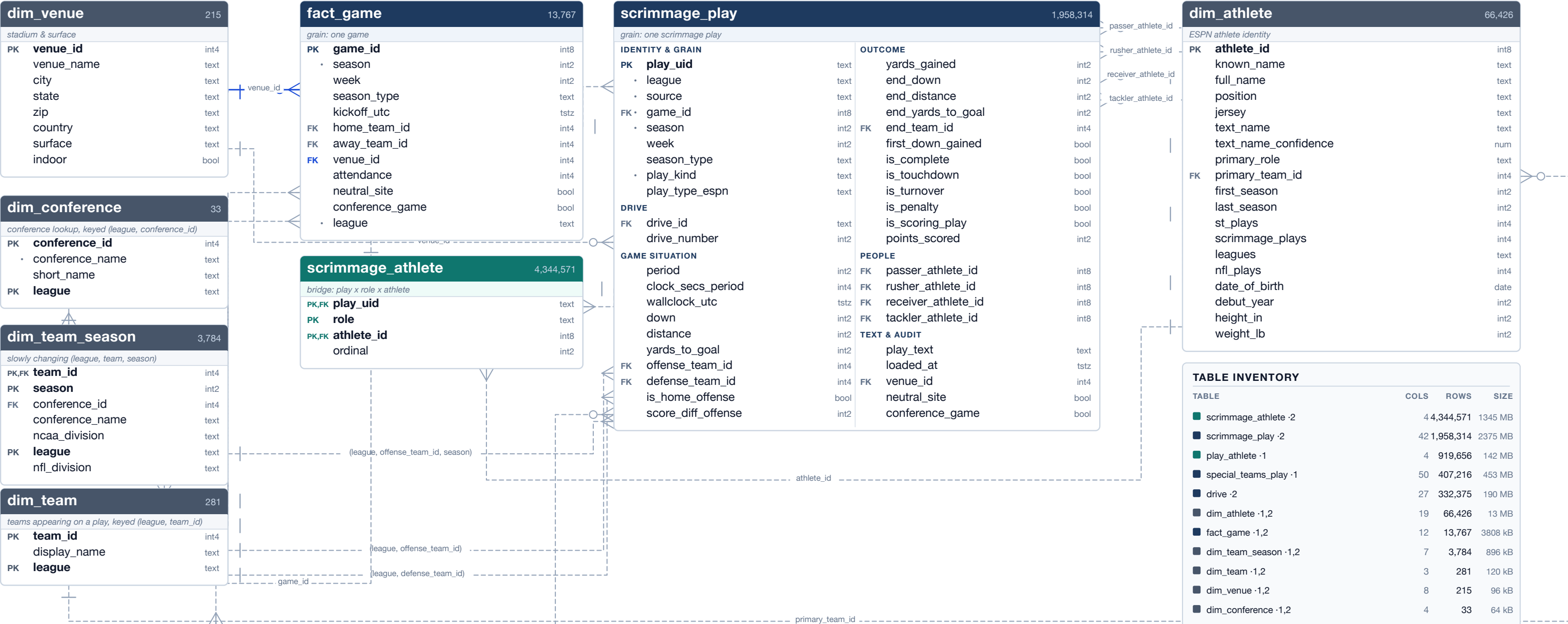


TABLE INVENTORY

TABLECOLSROWSIZE

scrimmage_athlete ·244,344,5711345 MB

scrimmage_play ·2421,958,3142375 MB

play_athlete ·14919,656142 MB

special_teams_play ·150407,216453 MB

drive ·227332,375190 MB

dim_athlete ·1,21966,42613 MB

fact_game ·1,21213,7673808 kB

dim_team_season ·1,273,784896 kB

dim_team ·1,23281120 kB

dim_venue ·1,2821596 kB

dim_conference ·1,243364 kB

RELATIONSHIP INVENTORY

cardinality and orphan counts measured live against the data

PARENT (ONE)CHILD (MANY)JOIN KEYOPTIONALENFORCEDORPHAN ROWS

dim_venuefact_gamevenue_idyesFK0

dim_venuescrimmage_playvenue_idyesbuild0

dim_conferencedim_team_season(league, conference_id)nobuild0

dim_teamdim_team_season(league, team_id)nobuild215

dim_team_seasonscrimmage_play(league, offense_team_id, season)nobuild0

dim_teamscrimmage_play(league, offense_team_id)nobuild0

dim_teamscrimmage_play(league, defense_team_id)nobuild0

dim_teamfact_game(league, home_team_id)nobuild0

dim_teamfact_game(league, away_team_id)nobuild0

fact_gamescrimmage_playgame_idnobuild0

fact_gamedrivegame_idnobuild0

drivescrimmage_playdrive_idyesbuild0

dim_athletescrimmage_playpasser_athlete_idyesbuild0

dim_athletescrimmage_playrusher_athlete_idyesbuild0

dim_athletescrimmage_playreceiver_athlete_idyesbuild0

dim_athletescrimmage_playtackler_athlete_idyesbuild0

dim_athletescrimmage_athleteathlete_idnobuild0

scrimmage_playscrimmage_athleteplay_uidnobuild0

dim_teamdim_athleteprimary_team_idyesbuild0

HOW TO READ THIS DIAGRAM

CARDINALITY (CROW'S FOOT)

exactly one

one or many

zero or many

LINE STYLE

declared FOREIGN KEY (1 of 16)

enforced by the build, not Postgres

COLUMN MARKS

PKprimary key

PK,FKprimary key that is also a foreign key

FKreferences another table

·NOT NULL

TABLE ROLE

fact

bridge

dimension

drive				332,375
grain: one drive — spans both facts				
IDENTITY		COUNTS		
PK	drive_uid	text	offensive_plays	int2
·	league	text	plays_total	int2
	drive_id	text	plays_scrimmage	int2
FK	game_id	int8	yards	int2
·	season	int2	time_elapsed_secs	int4
	week	int2		
	season_type	text	START	
·	drive_number	int2	start_period	int2
			start_clock_secs	int4
			start_yards_to_goal	int2
			start_text	text
			END	
			end_period	int2
	result	text	end_clock_secs	int4
	display_result	text	end_yards_to_goal	int2
	description	text	end_text	text
	is_score	bool		

The four dimensions on the left and dim_athlete are the SAME tables as on sheet 1, drawn in the same place — one warehouse, two facts. play_uid is unique across both facts, so they UNION cleanly; drive spans them, which is why plays_total counts kicks that live in special_teams_play.